EASY BOOK IN FRAMEWORK

PART 1





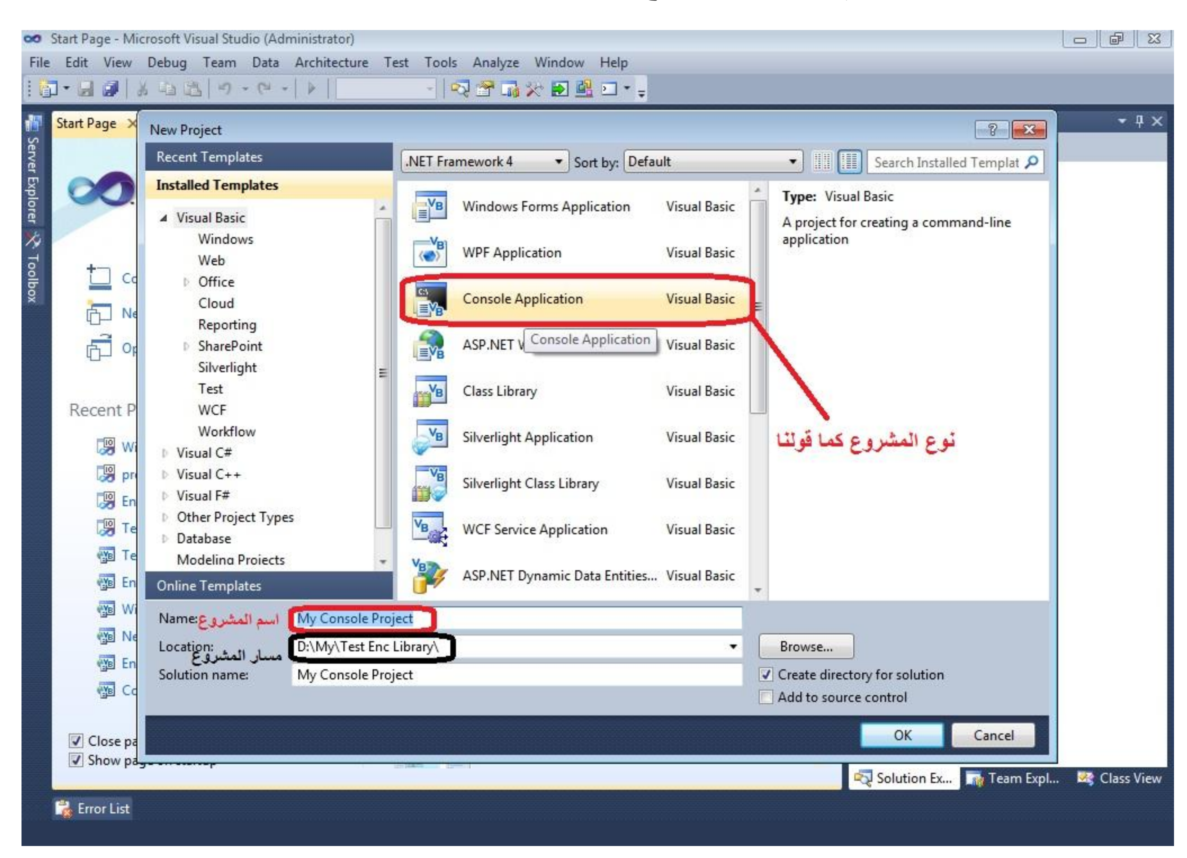
الوحده الاولى :-

ماهو الكونسول ابليكيشن ؟
CONSOLE APPLICATION

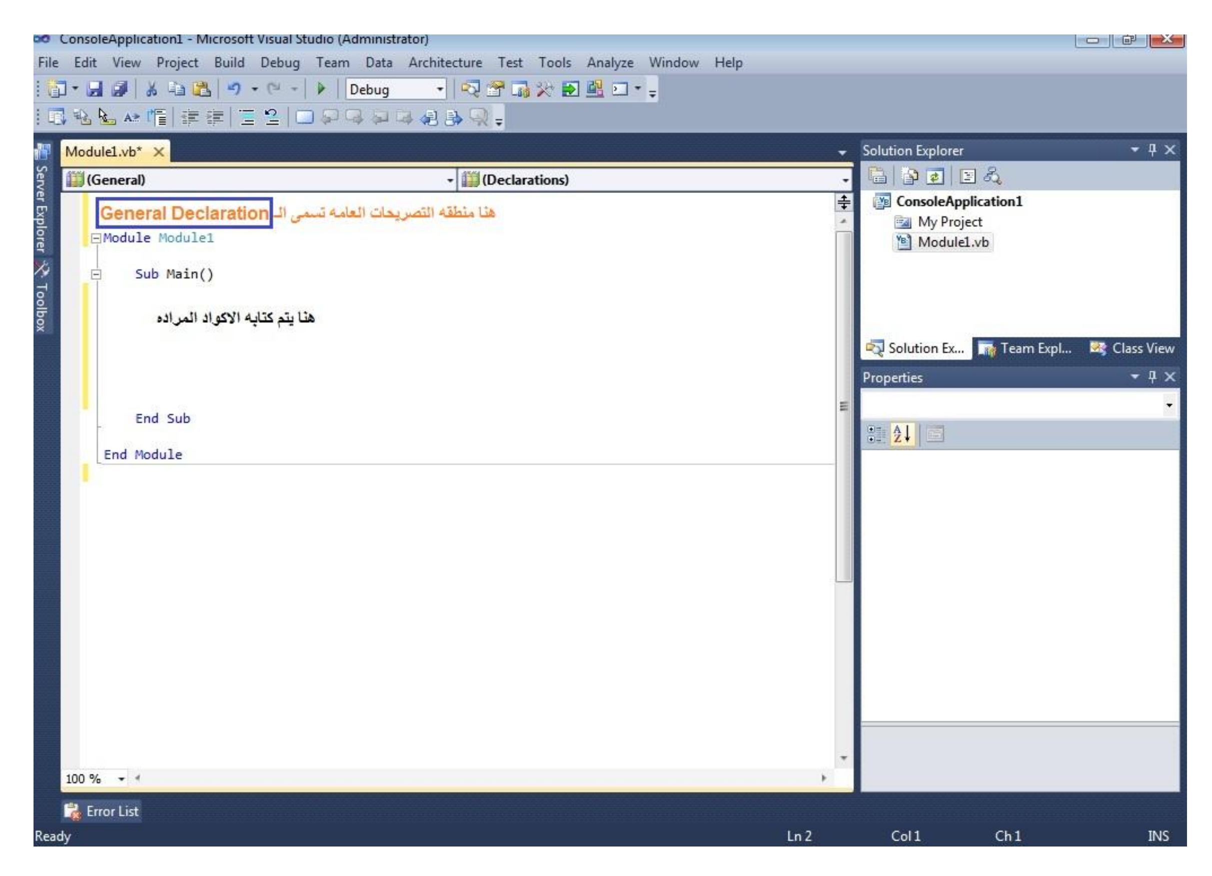
ان الحمدلله نحمده ونستعينه ونستهديه ونعوذ بالله من شرور انفسنا وسيئات اعمالنا انه من يهده الله فلا مضل له ومن يضلل فلا هادى له اما بعد :-
ماهو الــكونسول ابليكيشن ؟ يعتبر الكونسول من اهم التطبيقات نوعيا التى لايمكن الاستغناء عنها فى البرامج الكبيره وذلك لكبر اهميتها فى البرمجه تحت الدوس حيث انه يمكننا بها انتاج برامج متكامله تحت الدوس بالكونسول .

كيفيه انشاءه :-

قم بفتح الفيجول دوت نت ثم اختار مشروع جديد ثم Console Application ولا تنسى تحديد اسم ومسار المشروع على جهازك انظر الصوره التاليه :-



بعد الانتهاء من تحديد نوع واسم ومسار المشروع نقوم بالضغط على **Ok**
ليتم فتح المشروع >> ثم نأتى للتعرف على بعض النقاط قبل ان نبدأ فى شرح الاكواد
انظر الصوره التاليه :-



نلاحظ فى صفحه الاكواد تبدأ بـ

```
Module Module1
    Sub Main()
```

فهنا يكون موديول 1 هو الحاوى للصب الرئيسى فبدون **الموديول 1** لا يعمل المشروع فهذا هو جزء اساسى من المشروع
ايضا لا يمكن تغير اسم الصب الرئيسى واذا حدث به تغير تأكد ان المشروع سوف لن يعمل.
فكن على حذر من هذه الاخطاء البسيطه التى يقع بها الكثير ايضا تأكد ان نهايه صفحه الاكواد تنتهى بهاتان الجملتان على الترتيب

```
    End Sub
End Module
```

وتكتب الاكواد المراده تحت الصب الرئيسى كما ترى فى الصوره السابقه
ومن الممكن كتابه اكواد معينه فوق الموديل فى ماتسمى بمنطقه التصريحات العامه كماهو موضح بالصوره السابقه ليكون الكود معروفا فى المشروع كله ويمكنك استعماله تحت اى صب

مفاهيم :-
اخى العزيز لا بد ان تعرف بعض المفاهيم الاساسيه قبل ان نبدأ فى الكورس الخاص بالبرمجه تحت الدوس
من اهم ذلك المفاهيم (المخازن وانواعها وكيفيه حجزها)

انواع المخازن :-
المخازن لها انواع كثيره جدا. فى هذا الجزء لن نتحدث عليها ككل لانها لاتكفى للتحدث عنها فى كورس خاص ولكن سنأخذ اهم المخازن التى سوف نستخدمها فى هذا الكورس

المخزن النصى :-
وهو عباره عن مخزن يحمل بيانات من النوع النصى سواء كانت اسماء عملاء او ارقام تليفونات او ايميلات .... الخ فكل هذا يقع تحت مفهوم داتا نصيه

كيفيه حجزه

```
Dim M As String
```

المخزن العددى :-
وهو عباره عن مخزن يحمل بيانات من النوع العددى الصحيح فقط فلا يجوز ان نعطيه قيمه عدد عشرى

كيفيه حجزه

```
Dim M As Integer
```

المخزن العددى المزدوج:-
وهو عباره عن مخزن يحمل بيانات من النوع العددى الصحيح والمزدوج (اى الارقام العشريه)

كيفيه حجزه

```
Dim M As Double
```

المخزن بايت :-
وهو عباره عن مخزن يحمل القيم بايت التى هى جزء من اجزاء الملفات ووحده قياس لها وسوف نقوم بشرحه فى الوحدات القادمه وكيفيه استعماله فى تدمير الملفات

كيفيه حجزه

```
Dim M AS Byte
```

انتهت الوحده الاولى

الوحده الثانيه :-

بعض خصائص الكونسول

اعلم اخى القارىء اننى حينما اتحدث عن الكونسول اننى اتحدث عن شاشه الدوس التى سوف تظهر حين تنفيذ المشروع تابع شرح اكواد خصائص الكونسول بدقه وقم بتجربتها كلها لتشعر بالفرق

```
Console.Clear()        'يقوم هذا الكود بمسح شاشه الدوس ان كان بها كلام يفضل كتابته مع اول كل كود
Console.Beep()         'لاصدار صوت من شاشه الدوس
Console.BackgroundColor = ConsoleColor.Blue   'لتغيير لون ماخلف الكلام فى شاشه الدوس
Console.Title = ("Meko") 'لوضع عنوان معين فى شاشه الدوس والعنوانالافتراضى لها يكون على اسم المشروع
Console.ForegroundColor = ConsoleColor.DarkGreen  'لتغيير لون شاشه الدوس
Console.SetCursorPosition(10, 10)  'لتغيير مكان مؤشر البدايه
Console.Write("Welcome")           'للكتابه فى سطر قابل للزياده
Console.WriteLine("AGC Academy")   'للكتابه فى سطر مستقل لايمكن الكتابه فيه مره اخرى
Dim S As String = Console.ReadLine() 'هذا الكود لكى يقرأ الكونسول قيمه مايدخله المستخدم بالدوس
Console .ReadKey()  '    يقوم هذا الكود بتثبيت شاشه الدوس وعدم اغلاقها تلاقئ الا عند الضغط على اى زرار
```

هل قمت بتطبيق الاكواد السابقه بدقه ؟
هل فهمت ماهو دور كل كود ؟
اذا دعنا نقوم بأمثله عينيه تقوم بتطبيق اشياء معينه

المثال الأول :-

مثال يقوم بعرض بيانات شخص فى شاشه الدوس تلقائيا '

```
Dim Name As String = "Malek"
Dim Phone As String = "0100309505"
Dim Mail As String = "Captian.malek@gmail.com"

 'لاحظ الكود التالى لعرض بيانات الشخص بجوار بعضها فى سطر واحد
Console.WriteLine("Name :{0} Phone:{1} Mail:{2}", Name, Phone, Mail)
Console.WriteLine("---------------------------------------------------")
Console.ReadKey()
```

لقد قمنا فى هذا المثال بحجز ثلاث مخازن من النوع النصى باسماء معينه ووضعنا بكل مخزن بيان لشخص معين
ثم أمرنا شاشه الدوس ان تضع بيانات الشخص الموجوده فى كل مخزن فى سطر واحد
ثم امرنا شاشه الدوس بعمل خط هكذا ---------- تحت بيانات الشخص
ثم عدم اخفاء شاشه الدوس بعد تنفيذ المطلوب الا عند الضغط على اى زرار من الكيبورد

هل قمت بتطبيق المثال ؟
ركز كويس وانت بتطبق وافهم الكود بيقول ايه وفايدته ايه تعالى نطور الكود شويه فى المثال التانى ؟

المثال الثانى :-

مثال يوضح عمل الكونسول عن طريق ادخال المستخدم قيم معينه '

```
 Console.Write("Enter UR Real Name : ")
       Dim Rname As String = Console.ReadLine

       Console.Write("Enter UR Real Phone : ")
       Dim RPhone As String = Console.ReadLine
```

```
        Console.Write("Enter UR Email :  ")
        Dim Rmail As String = Console.ReadLine

        Console.WriteLine("The Information U Entred Is ")

        Console.Write("Name: {0} Phone: {1} Mail: {2}", Rname, RPhone, Rmail)

        Console.ReadKey()
```

هو نفس المثال السابق ولكن المستخدم نفسه هو الى قام بادخال بيانات الشخص افهم الكود بيقول ايه
امرنا الكونسول بكتابه هذه الجمله فى سطر قابل للزياده Enter UR Real Name
وقمنا بحجز مخزن نصى وامرناه بحمل القيمه التى سيدخلها المستخدم

امرنا الكونسول بكتابه هذه الجمله Enter UR Real Phone
وقمنا بحجز مخزن نصى وامرنا المخزن بحمل القيمه التى سيدخلها المستخدم

امرنا الكونسول بكتابه هذه الجمله على شاشه الدوس Enter UR Email
ثم قمنا بحجز متغير نصى ليحمل قيمه مايدخل المستخدم فى الشاشه

ثم امرنا الكونسو بكتابه هذه الجمله فى سطر مستقل The Information U Entred Is
ثم امرنا الكونسول بكتابه البيانات التى ادخلها المستخدم والتى كان يحملها كلا من المخازن الثلاثه السابقه

ثم امرنا الكونسول بعدم اغلاق شاشه الدوس فور تنفيذ الكود الا عند الضغط على اى زرار فى الكيبورد

ملحوظه مهمه جدا :-

من المعروف عند حجز مخزن من **Boolean** الذى يرجع بقيمه نعم او لا فانه عند حجزه يحمل القيمه لا
النوع
فاذا اردنا ان نقوم بحجزه بقيمه فارغه نقوم بكتابه كود الحجز كالاتى :-

```
        Dim A As Nullable(Of Boolean) = Nothing   ' لحجز مخزن يحمل قيمه خاليه
```

انتهت الوحده الثانيه :

اخى القارىء قبل البدء فى الوحده الثالثه اتمنى ان تكون قد طبقت الأمثله السابقه بدقه وفهم معنى كل كود لان الاتى مترتب على السابق

الوحده الثالثه :-

ARRAY (DYMENTION)

المصفوفات

تحمل المخصوصه صفات المخزن العادى ولكنها تحتوى على اجزاء دعنا نلاحظ الفرق مع الامثله

المثال الاول :-

' ده مثال مبسط يوضح كيفيه عمل المصفوفه

```
Dim B(2)
        B(1) = 200
        B(2) = 300

        Console.WriteLine(B(1))
        Console.WriteLine(B(2))
        Console.ReadKey()
```

فى المثال السابق قمنا بحجز مصفوفه تحتوى على جزئين
ثم وضعنا فى الجزء الاول القيمه 200
والجزء الثانى 300
ثم امرنا الكونسول بكتابه محتويات المخزن الاول وكتابه محتويات المخزن الثانى
هل لاحظت كيفيه عمل المصفوفه ؟ هل عرفت ما الفرق بينها وبين المخزن العادى ؟
اذا هيا بنا نذهب لنطور المثال اكثر لنتعرف على مميزات المصفوفات

المثال الثانى:-

' مثال يوضح كيفيه هدم المصفوفه القديمه واعاده بنائها من جديد لزياده اجزائها

```
Dim Z(2)
Z(1) = 500
Z(2) = 600
ReDim Z(3)
Z(1) = 5
Z(2) = 6
Z(3) = 7
Console.WriteLine(Z(1))
Console.WriteLine(Z(2))
```

```
        Console.WriteLine(Z(3)
        Console.ReadKey()
```

قمنا فى هذا المثال بحجز مصفوفه تحتوى على جزئين ووضعنا بكل جزء قيمه معينه
ثم قمنا باعاده بنائها من جديد لزياده اجزائها وجعلناها تحتوى على ثلاثه اجزاء ووضعنا بكل جزء قيم معينه غير القيم القديمه لانها تم هدمها
ثم أمرنا الكونسول بكتابه مافى أجزاء المصفوفه فى شاشه الدوس

المثال الثالث:-

' مثال يوضح كيفيه زياده عدد اجزاء المصفوفه بدون هدمها

```
Dim Q(2)
        Q(1) = 100
        Q(2) = 200
        ReDim Preserve Q(4)
        Q(3) = 300
        Q(4) = 400
        Console.WriteLine(Q(1))
        Console.WriteLine(Q(2))
        Console.WriteLine(Q(3))
        Console.WriteLine(Q(4))
        Console.ReadKey()
```

ركز فى شرح المثال ده عشان تفهمه كويس
قمنا بحجز مصفوفه تحتوى على جزئين ووضعنا بالجزء الاول والثانى قيم معينه
ثم قمنا بزياده اجزاء المصفوفه الى 4 اجزاء بدلا من 2
ثم قمنا بوضع قيم معينه فى الجزء الثالث والرابع
ثم امرنا الكونسول بكتابه محتويات الاربع اجزاء فى شاشه الدوس

هل قمت بتطبيق المثال ؟
هل فهمت عمل كل كود من اكواده ؟
اذا هيا بنا لنقوم بتطوير المثال اكثر لنرى امكانيات المصفوفات

المثال الرابع:-

هذ المثال يجعل المستخدم يقوم بانشاء المصفوفه بنفسه بحيث يدخل عدد اجزائها ويضع القيم لكل جزء'

```
Console.Write("Enter the Parts of Array : ")
        Dim C As Integer = Console.ReadLine
        Dim D()
        ReDim D(C) 'هل تذكر هذا الكود ؟ لاعاده بناء المصفوفه من جديد'
        Dim E As Integer = 0

        For E = 1 To D.Length - 1
            Console.Write("Enter The No. {0} : ", E)
            D(E) = Console.ReadLine

        Next

        Console.ReadKey()

        Console.WriteLine("The Numbers U entred is ")
        Console.WriteLine("------------------------")

        Dim F As Integer = 0
        For F = 0 To D.Length - 1
            Console.WriteLine(D(F))
        Next
        System.Array.Sort(D)  ' يقوم الكود بعمل فرز لعناصر المصفوفه وذلك بترتيب اجزاها من الكبير الى الصغير
        System.Array.Reverse(D)  'و هذا عكس السابق
        Console.ReadKey()
```

ركز فى المثال ده كويس جدا هتلاقى حجات جديده عليك
امرنا الكونسول بأن يسال المستخدم بادخال عدد اجزاء المصفوفه
ثم قمنا بحجز متغير من النوع العددى ليحمل قيمه ما ادخله المستخدم (عدد اجزاء المصفوفه)
ثم قمنا بحجز مصفوفه خاليه الاجزاء
ثم قمنا باعاده بنائها بعدد الاجزاء الذى قام بادخاله المستخدم (الذى كان يحمله المخزن العددى)
ثم قمنا بعمل دواره تقوم بعمل الاتى
تسأل المستخدم عن ادخال القيمه التى يريد وضعها فى كل جزء من الاجزاء
وتقوم بقراءه كل قيمه ووضعها فى الاجزاء على الترتيب

بعد ما قام المستخدم بوضع قيمه لكل جزء من اجزاء المصفوفه تنتظر الشاشه منه ان يضغط على اى زرار لتنفيذ الاتى :-
عمل سطر فاصل هكذا ---------------------
ثم عرض قيم اجزاء المصفوفه التى قام بادخالها المستخدم
ثم فرزها (ترتيبها من الرقم الكبير للصغير)
ثم فرزها مره اخرى (عكس السابق)

هل فهمت المثال جيدا ؟
هل فهمت عمل كل كود من اكواده ؟
نعم انه بالفعل صعب الى حد ما لانه يحتوى على اكواد جديده ولكن اذا قمت بتطبيقه اكثر من مره وقراءه الشرح بدقه سوف تتغلب على كل الصعوبات ثم هيا بنا لنطور المثال اكثر ؟

المثال الخامس:-

مثال يوضح كيفيه عمل بحث داخل مصفوفه عن طريق داله البينارى سيرش '

```
Dim M(2) As String
        M(1) = ("Ahmed")
        M(2) = ("Mohmaed")
        System.Array.Sort(M)  ' ملحوظه هامه جدا : لايمكن اتمام المثال الا بعمل فرز اولا للمصفوفه بالكود التالى

        Console.Write("Enter The Name U want To Find :  ")
        Dim H As String = Console.ReadLine
        Dim K As Integer = System.Array.BinarySearch(M, H)
        Console.WriteLine(K)
        Console.ReadKey()
```

فى هذا المثال قمنا بحجز مصفوفه من النوع النصى (اى يكتب فى اجزائها نصوص)
ثم قمنا باعطاء كل جزء من اجزائها قيمه معينه (احمد) و (محمد)
ثم قمنا بعمل فرز للمصفوفه وذلك ليتم ترتيب محتويات اجزائها واعلم انك اذا ما اضفت كود الفرز قبل كود البحث لن يفلح معك الكود
ثم امرنا شاشه الدوس ان تطلب من المستخدم ان يدخل الاسم المراد البحث عنه
ثم قمنا بحجز مخزن نصى ليحمل القيمه التى سوف يدخلها المستخدم
وقمنا ايضا بحجز مخزن عددى (أى يحمل بيانات من النوع العددي فقط بدون كسور) ليحمل ناتج جمله البحث
ثم أمرنا الكونسول بكتابه ناتج جمله البحث (الذى كان مخزون فى المخزن العددى)
اعلم اخى القارىء ان ناتج هذا المثال سيعطيك رقم الجزء الذى يحتوى على قيمه البحث
فمثلا ان كتبنا **Ahmed** فتكون ناتج البحث 1 اى هذه القيمه موجوده فى الجزء الاول للمصفوفه فى البحث

مثال بسيط جدا وسهل ولا يحتوى على تعقيدات
اذا هل قرأته بكل دقه ؟
هل فهمته بكل اكواده ؟
هل قمت بتطبيقه اكثر من مره ليصل الى عقلك ؟
اذا اعلم انك فهمت طريقه عمل المصفوفات وهكذا تكون قد انتهى درس المصفوفات
اتمنى ان يكون قد وصلت بفكرك الى ارقى المستويات

انتهت الوحده الثاثه:-

تذكره :-

{ وذكر فان الذكرى تنفع المؤمنين }

قبل ان تبدأ فى الوحده الرابعه ارجوووووووووووو ان تكون قد انتهيت من تطبيقات الوحدات السابقه لانه كما قولنا كل الامثله والشروحات القادمه مبنيه على السابقه

الوحده الرابعه:-

العمليات على النصوص فى الكونسول

تهدف هذه الوحده الى :-

شرح كيف يتعامل الكونسول مع النصوص من حيث عرض اول حرف من كلمه مخزنه
او من منتصفها او البحث عن حرف معين فى كلمه مخزنه .... الخ

```
Dim M As String = "Malek"
  Dim F As String = "Future Developer"
  Dim L As String = M + F

   Console.Write("Enter UR Name : ")
   Dim R As String = Console.ReadLine
   R = LSet(R, 3) 'يقوم الكود باقتطاع الكلمه الموجوده فى المخزن من اليسار ليجعل حروفها3 حروف فقط
   R = RSet(R, 3) 'يقوم الكود باقتطاع الكلمه الموجوده فى المخزن من اليمين ليجعل حروفها3 حروف فقط
   R = M & Space(5) & F   ' يقوم الكود بجمع مافى المخزنين بجوار بعضهم ويضع بينهم5 مسافات

   Console.Write(R)

   Console.Write(Left(R, 2)) 'لاظهار  حرفين من يسار الكلمه الموجوده فى المخزن
   Console.WriteLine(Right(R, 2)) ' لاظهار اول حرفين من يمين الكلمه الموجوده فى المخزن المحدد
   Console.WriteLine(InStr(R, "M"))  ' يقوم الكود باظهار رقم الحرف  M  الموجود فى الكلمه
   Console.WriteLine(Mid(R, 3, 1)) ' يقوم الكود التالى باظهار ثلاث حروف ابتداء من الحرف الاول فى الكلمه
   Console.WriteLine(Replace(R, "F", "M"))  ' يقوم بتبديل الحرف  F  مكان الحرف  M  فى الكلمه
   Console.ReadKey()
```

فى هذا المثال

قمنا بحجز ثلاث متغيرات من النوع النصى
ووضعنا بالمخزن الاول والثانى قيمه معينه
ثم جعلنا المخزن الثالث يساوى مافى المخزن الاول والمخزن الثانى
ثم امرنا الكونسول بسؤال المستخدم ان يكتب اسمه لنقوم ببعض العمليات عليه
فقمنا بهذه العمليات التى تم شرح كل كود من اكوادها فى المثال السابق
لا اعلم ماذا اقول لك من اهميتها !! ولكنك ستلاحظ كبر اهميه هذه العمليات حينما تحتك بالسوق والبرمجيات عمتا

انتهت الواحده الرابعه:-

الوحده الخامسه :-

STRING BUILDER

تهدف هذه الوحده الى :-

كيفيه التعامل مع النصوص ولكن عن طريق استخدام **String Builder**
وللعلم انه من الممكن ان نستغنى عن استعماله ونقوم باستعمال الطريقه العاديه كما اخذنا فى الوحده الرابعه ولكن هذا يرجع الى حجم الداتا ان كانت الداتا بأحجام صغيره فيمكننا استخدام الطريقه العاديه كما اخذنا فى الوحده الرابعه
.

اما ان كانت حجم الداتا كبيره تتراوح الى الجيجات ففى هذه الحاله لابد ان نستخدم الـ **String Builder**
وهذا يرجع الى سرعه تنقله بين الحروف والكلمات وهذا مايقوم المبرمجين باستخدامه فى البرامج اداره الشركات الكبرى مثل شركات الاتصالات والفنادق والمستشفيات العامه ..... الخ .
ايضا يستخدم فى جزء من تشفير الرسائل والكلمات والباسوردات والداتا المهمه .
تعالى بنا نتعرف على الـ **String Builder** من خلال الأمثله :-

المثال الاول:-

```
' مثال يوضح كيفيه التعامل مع الاسترنج بيلدر
Dim S As New Text.StringBuilder
S.Append("Agc")   'للكتابه داخل المخزن على سطر قابل للاضافه
S.AppendLine("Academy")  ' للكتابه داخل مخزن الاسترنج بيلدر على سطر مستقل
S.Remove(5, 3)  ' لمسح خمس حروف من الكلمه التى بالمخزن ابتداء من الحرف الثالث لها
S.Replace("A", "M")  ' لتبديل كل حروف M فى الكلمه الى الحرف A
Console.WriteLine(S.ToString)  ' يكتب هذا الكود لعرض كل محتويات المخزن من كلمات
Console.ReadKey()
```

يقوم هذا المثال بشرح كيفيه التعامل بالاسترنج بيلدر مع النصوص بدلا من الطريقه العاديه
فهنا قمنا بحجز مخزن من النوع Text.StringBuilder الذى يتعامل مع النصوص
ثم امرناه باضافه داخل المخزن كلمه معينه على سطر يقبل للاضافه مره اخرى
ثم ضفنا بالمخزن كلمه اخرى ولكن على سطر مستقل غير قابل لاضافه كلمات اخرى ليكون مافى المخزن هو

Agc
Academy

ثم امرنا المخزن بحذف اول خمس حروف من الكلمه التى به ابتداء من الحرف الثالث لها ليكون الناتج كالاتى

Ag
emy

ثم امرناه بتبديل حرف **M** ويضع مكانه حرف الـ **A** ليكون الناتج كالاتى

ثم امرنا بعرض كل العمليات التى تمت عليه على شاشه الكونسول (شاشه الدوس)

تعالى بنا نطور المثال اكتر ؟

المثال الثانى:-

تشفير الرسائل :
قبل ان نبدأ بالمثال عليك ان تعلم بعض النقاط المهمه
اننا عند كتابه اى رساله مهما كان محتوياتها من كلمات فان كلماتها تتكون من حروف .
فلابد ان تعلم ان لكل حرف من الحروف رقم معين تتعامل به لغات البرمجه كلها ويطلق عليها (اكواد الاسكى) فمثلا زرار الانتر فى الكيبورد رقمه 13 وهكذا .....
وتنحصر هذه الارقام من صفر حتى 255 تحتوى على كل العلامات والحروف الموجوده فى الكيبورد
ومنها الارقام المستخدم حروفها اكثر استخداما هى الارقام مابين 45 الى 122
دعنا مع هذا المثال البسيط نظهر مانقول

مثال بسيط يقوم بعرض اكواد كل الحروف والعلامات التى بالكيبورد '

```
Dim A As Integer = 0
        For A = 0 To 255
        Console.WriteLine(Chr(A))
        Next
        Console.WriteLine("--------")
        Console.WriteLine(Chr(13))
        Console.ReadKey()
```

هل توصلت الان الى ما أقول ؟
اذا دعنا نأخذ مثال على كيفيه تشفير الرسائل

' تشفير الرسائل

```
Dim SB As New Text.StringBuilder
        SB.Append("How R U")
        Dim E As Integer
        For E = 45 To 122
            SB.Replace(Chr(E), Chr(E + 125))
        Next
        Console.WriteLine(SB.ToString)
        Console.ReadKey()
```

فى هذا المثال قمنا بحجز مخزن من النوع Text.StringBuilder
ثم قمنا بوضع القيمه How R U فى المخزن
ثم قمنا بحجز مخزن من النوع العددى ليحميل اعداد لنستخدمها كأكواد لحروف المخزن
ثم قمنا بعمل دواره تقوم بالاتى
ان يبدا المخزن العددى من الرقم 54 وينتهى عند الرقم 125
امرنا مخزن الاسترنج بيلدر بان يقوم بتبديل كل الحروف التى كودها هو مافى المخزن العددى
مكان حرف يحتوى على كود ازيد منه ب 125 (ليقوم بتبديل حرف مكان حرف اخر لتصبح الرساله غير مفهومه)
ثم امرنا شاشه الكونسول بان تكتب محتويات المخزن استرنج بيلدر بعد التعديل
ليصبح الناتج كالأتى
د ز ô إى
انظر كيف اصبحت الجمله How R U مشفره !!

كيف نفك التشفير عن هذه الجمله ؟
تعالى نفكر سويا فى كيفيه فك التشفير عن هذه الجمله لتعود لطبيعتها
هل فكرت جيدا ؟؟؟ اعلم انك وصلت الى حل لانك قد فهمت المثال وطبقته اكثر من مره وفهمت كل اكواده
اجل هذا هو الحل بارك الله فيك

' فك التشفير عن الرساله

```
For E = 45 To 122
            SB.Replace(Chr(E + 125), Chr(E))
        Next
        Console.WriteLine(SB.ToString)
        Console.ReadKey()
```

يقوم هذا الكود بعمل دواره تقوم بعكس ماقامت به الدواره الاولى
لتعود الجمله لطبيعتها ويصبح الناتج كالاتى
How R U

ما رأيك فى طريقه التشفير التى يعتقد الكل انها معقده فى فهمها ؟
اعلم انها اسهل مما كنت تتصور
اذا تعالى بنا نفكر سويا فى ايجاد طريقه لتعقيد التشفير اكثر ليصعب على المستخدم فك تشفيرها فى المثال القادم

المثال الثالث:-

مثال يوضح تشفير الرسائل بطريقه اقوى من السابق لتصبح رساله معقده التشفير '

```
Dim Sbb As New Text.StringBuilder
        Console.Write("Enter The MSg U like To Encrypt : ")
        Dim G As String = Console.ReadLine
        Sbb.AppendLine(G)
        Dim H As Integer = 0
        For H = 45 To 122
            Sbb.Replace(Chr(H), Chr(H + 100) & "@!#")
        Next

        Console.Write("The Msg After Encrypt : {0}", Sbb.ToString)
        Console.ReadKey()
```

فى هذا المثال قمنا بحجز مخزن من النوع **Text.StringBuilder**
وامرنا الكونسول ان يسأل المستخدم عن الكلمه التى يريد تشفيرها
وقمنا بحجز مخزن من النوع النصى لكى يحمل قيمه ماكتبه المستخدم
ثم امرنا مخزن الاسترنج بيلدر ان يقوم باضافه ماكتبه المستخدم اليه ليقوم بعمل التشفير عليه
ثم قمنا بحجز مخزن عددى لعمل دواره عليه كما فى المثال السابق
هذه الدواره تقوم بتبديل الحروف مكان حروف اخرى وتضيف مع كل حرف هذه الكلمات المعقده **@!#** لتصبح الرساله شديده التعقيد
فاذا كان قيمه ما كتبه المستخدم هو How R U
فيكون الناتج كالاتى
?@!#غ#!@س@!# ?@!# ?@!#
انظر كيف اصبحت الكلمه معقده بعد التشفير !!!!
قم بتجربتها واستنتج بنفسك !!

اعلم انك الان تفكر فى كيفيه فك التشفير عن هذه الجمله صعبه التعقيد
واعلم ايضا ان الموضوع شيق لك
اخى القارىء نعم هو نفس فكره كود فك التشفير السابق

```
'فك التشفير
For H = 45 To 122
  Sbb.Replace(Chr(H + 100) & "@!#", Chr(H))
Next
  Console.Write("The Msg After Decrypt : {0}", Sbb.ToString)
  Console.ReadKey()
```

يقوم الكود الاتى بعمل دواره تقوم بعكس ماقامت به دواره التشفير
لتقوم باعاده الكلمه الى طبيعتها مره اخرى . ليكون الناتج كالاتى
How R U

انتهت الوحده الخامسه :-

الوحده السادسه:-

اتصال الكونسول بقواعد البيانات

تهدف هذه الوحده الى :-
عن طريق مكتبه معرفه كيفيه انشاء اتصال بين الكونسول وقواعد البيانات عن طريق الادو
قبل ان نبدأ فى عمل اتصال بين اى قاعده البيانات والكونسول لابد ان نقوم بحجز مكتبه الادو وذلك عن طريق اتباع الخطوات الاتيه :-
من قائمه الادوات نختار
Project >> Add Reference… >> Com ثم من المنفذ >> Microsoft ActiveX Data Object 2.8 Library

يبقى لدينا ان نحصل على كود جمله الاتصال الذى سيكتب لنقوم بتوصيل قاعده البيانات بالكونسول
.
من فضلك لاتذهب بعقلك الى تفكير اخر شدد تفكيرك معى بقدر ماتستطيع لتعلم كيف تحصل على كود اتصال لاى قاعده بيانات بصرف النظر عن نوعها

اتبع الخطوات الاتيه :-

فى مجلد المشروع ثم قم بفتحه ومن قائمه الادوات له اختار حفظ بأسم ثم اختار اى اسم UDL
قم بانشا ملف TxT وضف له الامتداد.
كما هو موضح بالصور التاليه :-

Name | Date modified | Type | Size

DataBase — ٢٠١١/٠٨/٢١ ص ٠٢:١٣ — File folder
project — ٢٠١١/٠٨/١١ م ٠٣:٥٠
MyDatabase — ٢٠١١/٠٩/٠٦ ص ٠٦:٠
project — ٢٠١١/٠٨/٠١ م ٠٦:٤
project — ٢٠١١/٠٩/٠٦ ص ٠٣:٠

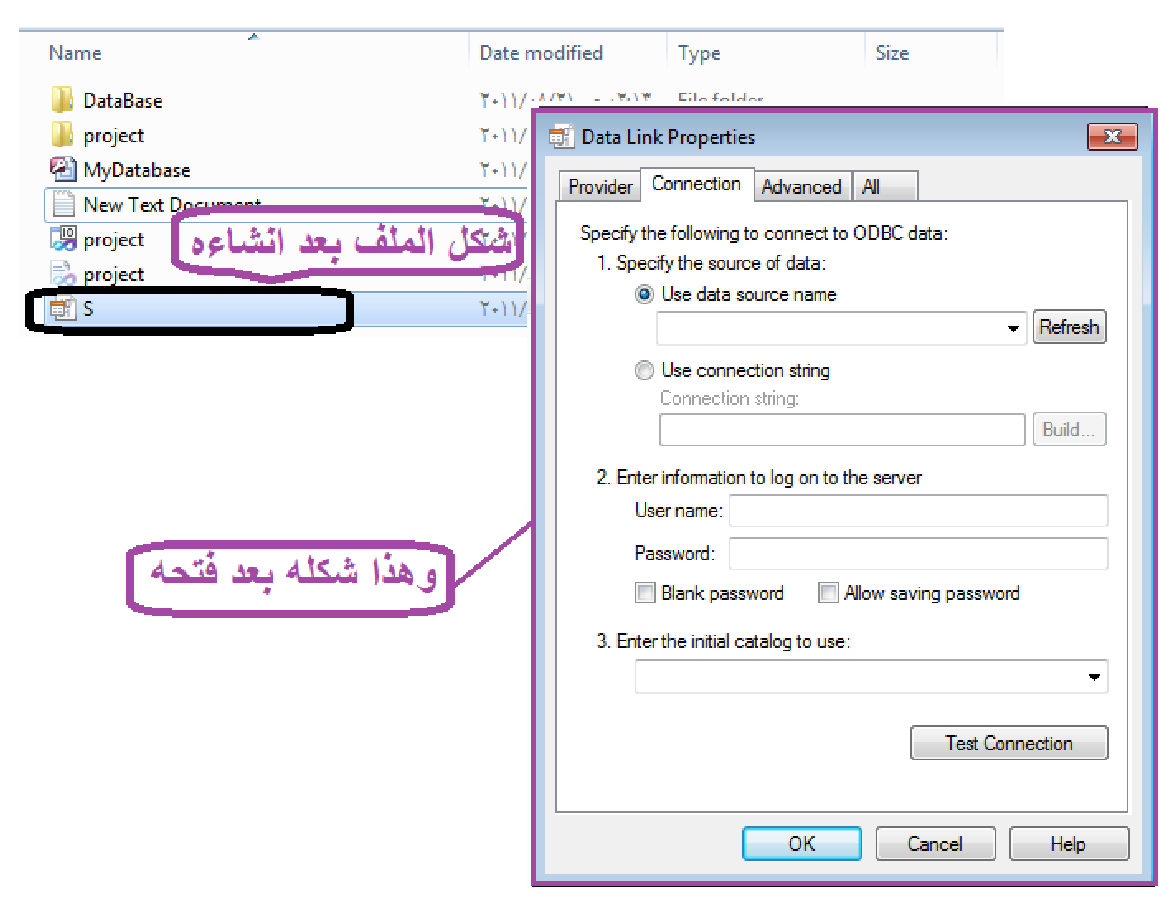
1

View
Sort by
Group by
Refresh
Customize this folder...
Paste
Paste shortcut
Undo Delete Ctrl+Z
Share with
Groove Folder Synchronization
New
Properties

Folder
Shortcut
Microsoft Office Access 2007 قاعدة بيانات
Bitmap image
Skin Builder Style Project
Contact
Microsoft Office Word Document
Journal Document
Microsoft Office PowerPoint Presentation
Microsoft Office Publisher Document
WinRAR أرشيف
Setup Factory 7.0 Document
Skin Builder Project
ATL MFC Trace Tool settings file
Text Document
Windows Live Call
Codejock Command Bars Designer Document
Microsoft Office Excel Worksheet
WinRAR ZIP أرشيف
Briefcase

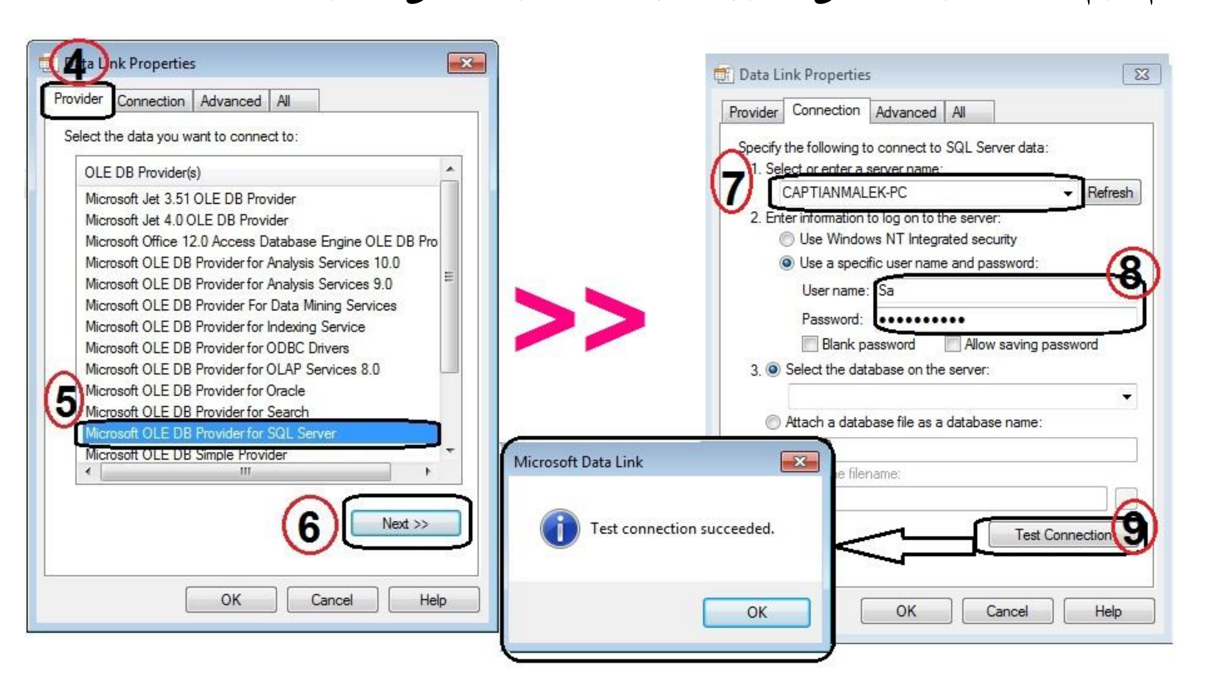


ليصبح لديك ملف بالشكل كما هو موضح بالصوره التاليه



كما ترى ايضا شكل الملف بعد فتحه

ثم تقوم بعمل الخطوات كما فى الصوره الاتيه من الخطوه 4 حتى الخطوه 9



الخطوه 1 : تحديد التبويب الذى ستبدأ اختيار نوع قاعده البيانات منه

الخطوه 5: تقوم بتحديد نوع قاعده البيانات المراد الحصول على كود الاتصال لها (فى الصوره تم تحديد نوع قاعده البيانات سيكوال سيرفر).

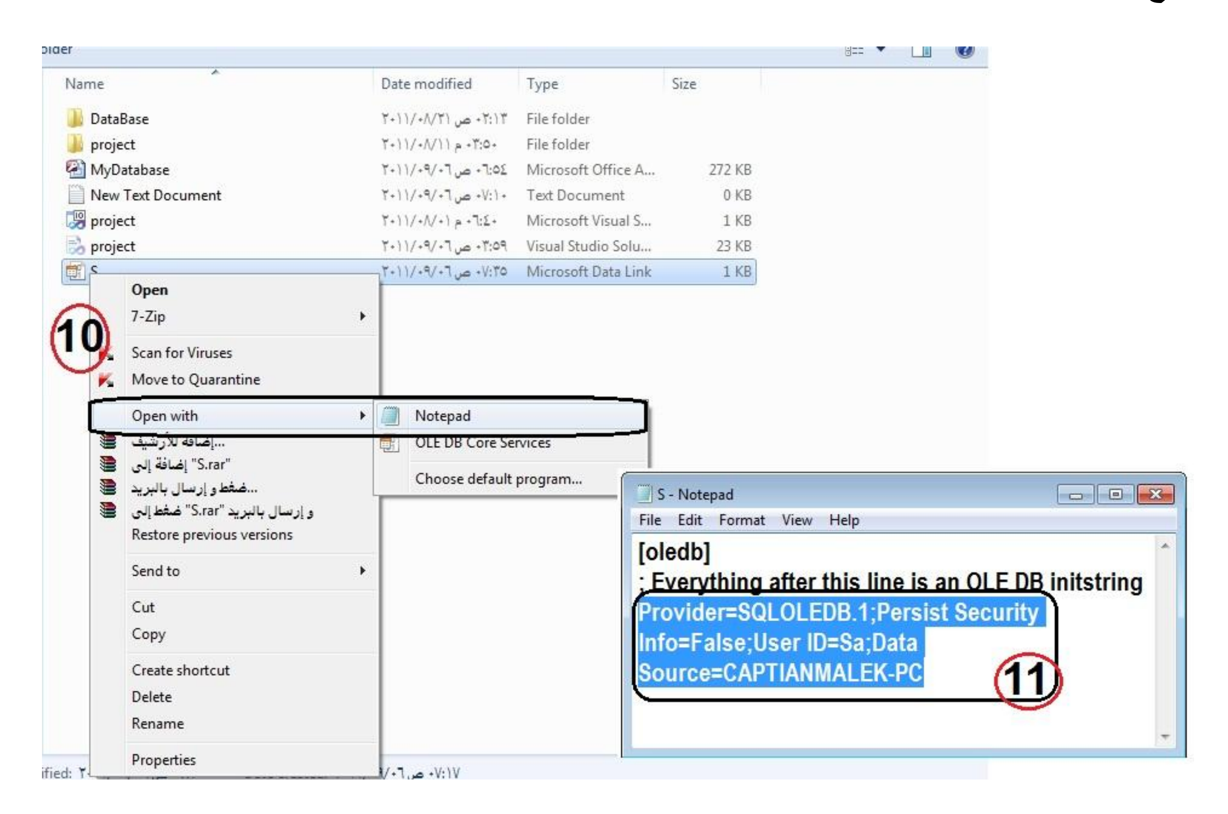
الخطوه 6 : اضغط على `Next` لينتقل بك الى التبويب الاخر

الخطوه 7 : قم بكتابه رابط او اى بى السيرفر الذى يحتوى على قاعده البيانات وان كانت على جهازك فاكتب اسم جهازك

الخطوه 8 : اكتب اسم وكلمه سر المستخدم لقاعده البيانات ان وجد

الخطوه 9 : عمل اختبار للاتصال ليعطيك رساله تفيد بنجاح الاتصال

هكذا قم انتهينا من عمل الاتصال بنجاح ولكن الى الان لم نحصل على كود الاتصال !!!
قم بفتح الملف عن طريق النوت باد ستجد به جمله الاتصال مايهمنا فى هذه الجمله
هو من اول الكلمه Provider حتى نهايه الكود
تابع الصوره التاليه :-



الخطوه 10 : لفتح الملف عن طريق النوت باد للحصول على كود الاتصال
الخطوه 11 : لتحديد كود الاتصال وعمل نسخه منه لنضعها فى كود الاتصال لنقوم بعمل التوصيل بين القاعده والكونسول

بعد تنفيذ كل هذه الخطوات
تعالى بنا نأخذ مثال لعمل اتصال بين قاعده بيانات **SQL Serever** والكونسول ابليكيشن

```
Module module1
    Dim cn As New ADODB.Connection
    Dim RS As New ADODB.Recordset

 Sub main()
        cn.Open("provider=
sqloledb;database=AGc;Uid=Sa;Pwd=0100309505;server(local)")
        RS.Open("Student", cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockBatchOptimistic)



    End Sub

End Module
```

قام المثال السابق
بحجز مخزنين من مكتبه الادو تحت الموديول مباشره وذلك لتعميم المخزنين على مستوى المشروع لكى نستطيع المخزن التعامل معهم تحت اى صب االاول من النوع كونيكشن الذى يقوم بالاتصال مع قاعده البيانات
والثانى من النوع ريكورد ست الذى يقوم بفتح الجداول

انتهت الوحده السادسه :-

الوحده السابعه :-

FUNCATION

تهدف هذه الوحده الى :-

معرفه ماهو الفانكشن وكيفيه التعامل معه وكيف نستفاد منه؟

الفانكشن :

هو مثل الصب تماما يخزن به اكواد معينه ولكن الفرق انه يعود بقيمه معينه
ويستعدى تحت اى صب ليعود بالقيمه المرسله اليه
وينشأ الفانكشن تحت الموديول وليس بداخل الصب
دعنا نرى الامثله ليتضح لنا اكثر ماهو الفانكشن

المثال الأول:-

' مثال بسيط يوضح كيفيه عمل الفانكشن

```
Module module1

   Function Avg(ByVal X As Double, ByVal Y As Double, ByVal Z As Double) As Double

        Return (x + y + z) / 3

  End Function

    Sub main()

        Console.WriteLine(Avg(10, 20, 30))
        Console.ReadKey()

    End Sub



End Module
```

كما ترى فى هذا المثال انشئنا فانكشن يرجع بقيمه المتوسط الحسابى لثلاث ارقام
قمنا بانشا فانكشن تحت الموديول بالاسم **Avg** يحتوى على ثلاث مخازن
X من النوع الـ **Double** الذى يحمل ارقام صحيحه وعشريه ويعود ايضا الفانكشن بقيمه من النوع **Double**
Z, Y ,
ويقوم بعمل الاتى يجمع مافى المخازن الثلاثه ثم يقسمه على ثلاثه ليخرج المتوسط الحسابى لهم
ثم قمنا باستخدام هذا الفانكشن تحت الصب الرئيسى **Sub Main**
واعطيناه القيم المدوونه فى الاكواد ليرجع بقيمه المتوسط الحسابى للقيم الثلاث

هل فهمت مامعنى الفانكشن الان ؟
اذا تعالى بنا نطور من طريقه التعامل مع الفانكشن فى المثال القادم

المثال الثانى:-

مثال لتوضيح كيفيه تعامل الفانكشن مع الدواره'

```
Module module1

   Function Avg(ByVal x As Double, ByVal y As Double, ByVal z As Double) As Double

       Return (x + y + z) / 3

   End Function

Sub main()
        Dim M(3) As Double
        Dim H As Double = 0

        For H = 1 To M.Length - 1
            Console.WriteLine("Enter The three No {0} : ", H)
            M(H) = Console.ReadLine
        Next
        Dim L As Double = Avg(M(1), M(2), M(3))
        Console.WriteLine(L)
        Console.ReadKey()

    End Sub

End Module
```

فى المثال السابق

قمنا بعمل فانكشن يحتوى على ثلاث معاملات ويقوم بجمعهم وقسمتهم على عددهم ليخرج المتوسط الحسابى لهم
ثم قمنا باستخدام هذا الفانكشن تحت الصب الرئيسى ولكن عن طريق كتابه المستخدم للثلاث ارقام المراد اخراج المتوسط الحسابى لهم
وتم ذلك عن طريق استخدام مصفوفه تحتوى على ثلاث اجزاء و دواره تقوم بعمل الاتى
تعمل حتى نهايه اجزاء المصفوفه -1 فلماذا تكون -1؟
لان المصفوفات عموما تبدأ من الصفر فان كانت المصفوفه تحتوى على ثلاث اجزاء فهى تكون
صفر > 1<2 <3
بالتالى اصبح اربع اجزاء بالتالى سوف يحدث خطأ عند التنفيذ فلا بد من اضافه -1 بجوار اجزاء المصفوفه كما ترى فى المثال السابق
وتقوم الدواره بسؤال المستخدم عن الثلاث قيم حتى نهايه اجزاء المصفوفه
ثم تقوم بوضع الثلاث قيم المدخله .. فى اجزاء المصفوفه لتصبح المصفوفه محمله بثلاث قيم
ثم قمنا باستخدام الفانكشن على الثلاث القيم التى فى اجزاء المصفوفه ووضعنا قيمه المتوسط الحسابى فى مخزن L
ثم امرنا الكونسول بعرض مافى المخزن L (المتوسط الحسابى للثلاث ارقام المدخله)

المثال الثالث:-

OVERLOAD

المقصود بالـ OverLoad :
عندما نقوم بعمل فانكشن مثلا يقوم باخراج المتوسط الحسابى لثلاث ارقام ونفترض ان المستخدم يريد عمل متوسط حسابى لرقمين وثلاث ارقام فى نفس الوقت ؟ فما الحل ؟
الحل هنا هو استخدام OverLoad
وهو اضافه فانكشن اخر بنفس الاسم وبنفس الوظيفه ولكن يحتوى على 2 معامل فقط وليس 3 فعندما يدخل المستخدم قيمتين يعمل على الفانكشن الذى يحتوى على معاملين وعندما يدخل المستخدم ثلاث قيم يعمل على الفانكشن الذى يحتوى على ثلاث قيم انظر المثال القادم لتعرف اكثر ماهو المقصود الاوفر لود

مثال على الاوفر لود OverLoad'

```
Module module1

Function Avg(ByVal x1 As Double, ByVal x2 As Double, ByVal x3 As Double) As Double
        Return (x1 + x2 + x3) / 3
    End Function

    Function Avg(ByVal x1 As Double, ByVal x2 As Double) As Double
        Return (x1 + x2) / 2
    End Function


    Sub main()
        'فيمكننا الان ادخال قيمتين كما ترى
        Console.WriteLine(Avg(10, 2))
        Console.ReadKey()

        'او  ثلاث قيم
        Console.WriteLine(Avg(10, 20, 30))
        Console.ReadKey()
    End Sub

End Module
```

فى المثال السابق
كما ترى قمنا بانشاء 2 فانكشن بنفس الاسم AVg
ولكن احداهم يحتوى على 2 مخزن
والاخر يحتوى على 3 مخازن

وقمنا بتوضيح استخدامهم تحت الصب الرئيسى
فيمكنك الان كتابه قيمتين او ثلاثه حسبما تريد
ومن الممكن ايضا استعمال الاثنتين مع بعضهم البعض

تنبيه!!!

اخى القارىء هل فكرت فى معنى الموجوده بجوار كل مخزن من مخازن الفانكشن **ByVal**
هل تعلم انه يمكنك تغير هذه الكلمه الى **Byref** ؟
فهذه تحدد نوع المخزن ولكل منهم معنى وعمل معين

Byval :
وهى التى قمنا باستخدامها فى كل الامثله السابقه والتى لاتقبل التحديث على قيم المخازن فهى تعود به كمان كان

ByRef :
فهى تقبل التحديثات على معاملتها ومخازنها وسوف نقوم بعرض اهميه استخدامها فى المثال القادم

' مثال يوضح عمل ByRef التى تقبل التحديثات بعكس ال ByVal

```
Module module1

    Public Sub F(ByRef X As String)

        Dim SB As New Text.StringBuilder
        SB.Append(X)
        SB.Replace("http:\\", "")
        X = SB.ToString
    End Sub

    Sub main()

        Console.Write("Enter URL : ")

        Dim N As String
        N = Console.ReadLine
        F(N)
        Console.WriteLine(N)
        Console.ReadKey()

    End Sub


End Module
```

فى المثال السابق:-
تأمل اخى القارىء فى الاكواد
لقد قمنا بانشاء صب عمومى بالاسم **F** يحتوى على مخزن X من النوع الذى يقبل التحديثات **ByRef**

يقوم هذا الصب بعمل الاتى
ان كان عنوان الموقع يحتوى على القيمه **http://** يقوم بالتحديث عليه ويجعل العنوان بدونها
ثم امرنا الكونسول تحت الصب الرئيسى
ان يسال المستخدم عن كتابه عنوان موقعه مثلا فان كان المستخدم قد ادخل العنوان
http://www.AGCACAdemy.com
فيقوم الصب بالتحديث على العنوان وجعله كالاتى
www.AGCACAdemy.com

عزيزي القارىء قم بتجربه ان تجعل قيمه المخزن فى الصب العمومى من النوع **ByVal** ولاحظ النتيجه

اعتقدت انك قد توصلت الى معنى كلا منهم الأن !!!
انتهت الوحده السابعه

الوحده الثامنه:-

EXCEPTION

التغلب على الأخطاء فى البرامج

تهدف الوحده الى :-
كيفيه التغلب على الاخطاء فى البرامج وكيف نتعامل مع البرنامج اذا حدث خطأ من المستخدم
وكيفيه التعامل اذا حدث خطأ برمجى غير ملحوظ
وذلك من خلال استخدام الداله **Exception** والداله **Err**
ركز فى المثال القادم سوف اقوم بشرح بعض مزايا الـدالتين وكيفيه استخدامهم

المثال الاول :-

مثال بسيط يوضح كيفيه التغلب على الاخطاء '
فى هذا المثال سوف اتعمد ان اخطأ فى الكود لتلاحظ كيف نتغلب على الخطأ '
سوف اقوم بحجز مخزن من النوع العددى وادخل به قيمه نصيه '

```
Try

Console.WriteLine("Enter Ur Name : ")
Dim M As Integer = Console.ReadLine

Catch ex As Exception

Console.WriteLine("Erorr Because U Dim M as Integer And Enter Name As string Pls
Redim M as string ")

'Exception مزايا استخدامات الداله
 Console.WriteLine("The Error Msg : {0}", ex.Message) 'لاظهار رساله الخطأ الحقيقيه
 Console.WriteLine("The Error Number : {0}", Err.Number) 'لمعرفه رقم الخطأ
Console.WriteLine("The Error Line : {0}", Err.Erl)      'لمعرفه رقم سطر الكود المحتوى على الخطأ

'Err مزايا استخدامات الداله
Console.WriteLine("The Error Msg : {0}", Err.Description) 'لاظهار رساله الخطأ
Console.WriteLine("The GetExceptoin : {0}", Err.GetException) 'للحصول على تفاصيل الخطأ
Console.WriteLine("The HelpContext : {0}", Err.HelpContext) 'للحصول على سياق المساعده
Console.WriteLine("The Error source : {0}", Err.Source) 'للحصول على مصدر الخطأ
Console.ReadKey()

End Try
```

ملاحظات هامه جدا!!!!!!!!!!

عند الشعور بحدوث خطأ فى كود معين نكتبه ونريد استخدام الداله Exception فلا بد ان تراعى التالى:-
تكتب فى بدايه الكود **Try**
ثم تكتب الكود المراد
ثم تكتب **Catch ex As Exception**
ثم تقوم بكتابه ماتريد ظهوره عند حدوث الخطأ ان كانت رساله خطأ او غيره ....
ثم تنهى الكود بـ **End Try**

فى المثال السابق :-
قمنا بكتابه Try فى اول الكود
ثم امرنا الكونسول بسؤال المستخدم عن الاسم
وقمنا بحجز مخزن من النوع العددى وامرناه بحمل القيمه التى يدخلها المستخدم
اذا سوف يحمل المخزن اسم المستخدم ولكن الاسم عباره عن نص والمخزن من النوع العددى فعند اذا لابد ان يقف البرنامج لحدوث هذا الخطأ فلقد قمنا بالتغلب عليه كما تر بالمثال
قمنا بكتابه Catch ex As Exception
ثم امرنا الكونسول ان يقوم باظهار بعد الرسائل كما شرحنا فى الاكواد
ثم قمنا بانهاء الكود بـ End Try

المثال الثانى :-

كيفيه توليد الاخطاء بالداله Err'

```
Console.WriteLine("Enter UR Name :")
Dim l As String = Console.ReadLine
Console.ReadKey()
Err.Raise(8)   'لتوليد الخطأ الذى رقمه 8 وايقاف البرنامج عن العمل
```

-اعلم اخى العزيز
انه يوجد ملايين الاخطاء المحتمله وكل هذه الاخطاء مسجله فى الفيجوال ولكل خطأ رقم معين

فى المثال السابق
امرنا الكونسول بسؤال المستخدم عن الاسم
وقمنا بحجز مخزن نصى وامرنا بحمل قيمه مايدخل المستخدم
ثم امرنا البرنامج بالوقوف واظهار الخطأ الذى رقمه 8 المسجل فى الفيجوال

هل تشعر بأن المثال السابق ليس له قيمه فى استخدامه فى البرمجه ؟
نعم اعلم ماتفكر به الان !!! ولكن دعنى اوضح لك ماهى اهميته فى المثال القادم

المثال الثالث :-

فكره عمل دواره للباسورد'

```
Begin:
        Console.Write("Enter UR PassWord : ")
        Dim M As Integer = Console.ReadLine
        Dim C As Integer
        If C > 3 Then Err.Raise(8)
        If M <> "123" Then

                Console.WriteLine("invaild Password")
                C = C + 1
                GoTo Begin


        End If
        Console.WriteLine("Welcome U have Enter Right PassWord ")
        Console.ReadKey()
```

فى المثال السابق قمنا بعمل طريقه واقعيه
فى اكثر البرامج و هو عند ادخال المستخدم كلمه السر اكثر من ثلاث مرات بطريقه خاطئه نقوم بتفجير خطأ وايقاف البرنامج .
اعتقد انك توصلت الى اهميه تفجير الاخطاء الان !! اذا استمع الى شرح الاكواد بدقه

قمنا بعمل عنوان فى بدايه الكود بالاسم **Bigen** ستعرف الان ماهى اهميه هذا العنوان
ثم امرنا الكونسول ان يطلب من المستخدم ادخال كلمه السر وقمنا بحجز مخزن عددى يحمل القيمه المدخله
ثم قمنا بحجز مخزن عددى ليقوم بحساب عدد المحاولات التى ادخل المستخدم فيها كلمه السر بطريقه خاطئه
ثم قمنا بعمل جمله شرطيه تنص على (ان كان قيمه هذا المخزن العددى اكثر من ثلاث محاولات قم بتفجير خطأ)
ثم قمنا بعمل جمله شرطيه اخرى تنص على (ان كان قيمه كلمه السر المدخله غير هذه القيمه 123 اذا)
اجعل الكونسول يفيد المستخدم بان كلمه السر خطأ ثم يقوم بزياده المخزن العددى الى 1 (المحتوى على عدد المحاولات) لحساب المحاوله
ثم يقوم باعاده الكود مره اخرى من بدايه العنوان Begin
ليقوم بسؤال المستخدم مره اخرى عن كلمه السر ...... وهكذا
فان ادخل المستخدم قيمه كلمه السر 123 اجعل الكونسول يعرض له رساله تفيد انه ادخل الباسورد بطريقه صحيحه
("Welcome U have Enter Right PassWord ")

هل توصلت الى فائده اضافه عنوان لكود معين مثل Begin فى المثال السابق؟
هل توصلت الى اهميه تفجير الاخطاء ؟
هل لاحظت فكره كيف تقوم بعمل دواره للباسورد ؟

انتهت الوحده الثامنه :-

الوحده التاسعه :-

INPUT OUTPUT

تهدف هذه الوحده :-
الى شرح كيفيه التعامل مع دوال الـ **InputOutput** التى تتعامل مع الملفات داخل الويندوز
ومن الممكن ان نختصر اسم هذه الدوال الى **IO**
وتتعامل هذه الدوال مع الملفات والمجلدات من حيث انشاء ملجدات وانشاء ملفات وعمل نسخ للملفات والمجلدات وتبديل مجلد مكان مجلد اخر .... الخ
أيضا كيفيه تشفير ملف معين وجعله خاص بمستخدم واحد فى الويندوز وكيفيه فك هذا التشفير وهذا عن طريق بعض الدوال.

ايضا انشاء ملفات تكست والكتابه بدخلها
ايضا تدمير جميع انواع الملفات سواء كانت صور فيديو برامج... الخ
واستعاده هذه الملفات الى طبيعتها مره اخرى وهى تعتبر ايضا من وسائل التشفير .

الداله **DirectoryInfo :**
تقوم هذه الداله بالتعامل مع المجلدات من حيث اظهار اسماء المجلدات ووقت انشائها ووقت التعديل عليها من حيث نسخها او تعديل اسمها .... الخ.
وتقوم ايضا بعرض محتويات مجلد او بارتشن معين من حيث المجلدات الموجوده به وتعرض مسار كل مجلد واخر مره تم الدخول عليه .
ايضا تقوم بانشاء مجلد وحذفه ونقله من مكان الى مكان اخر.

الداله **FileInfo :**
تستخدم هذه الداله لعرض الملفات التى بداخل مجلدات الداله **DirectoryInfo**
تقوم بعرض اسماء ملفاتها ومسار كل ملف وامتداد كلا منهم وحاله هذا الملف ان كان الملف مخفى او ظاهر للقراءه فقط .
ايضا يمكننا بهذه الداله تشفير ملف معين عن باقى مستخدمين الويندوز ليصبح هذا الملف صالح للمستخدم الحالى فقط ومشفر عن باقى المستخدمين .

الداله **:SreamWriter**
تستخدم هذه الداله للكتابه داخل ملفات التكست

الداله **StreamReader**
تستخدم هذه الداله لقراءه مابداخل اى ملف تكست

الداله : FileStream
يمكننا استخدمها فى تدمير جميع انواع الملفات سواء كانت صوره او فيديو او برنامج ...الخ
وايضا يمكننا اعاده ذلك الملفات الى طبيعتها مره اخرى

دعنا نذهب الى الامثله لترى كيفيه العمل مع دوال الـ **InputOutput**
وماهى خصائص ومميزات هذه الدوال .

المثال الاول:-

مثال يوضح كيفيه التعامل مع المجلدات من خلال دوال InputOutput'

```
Imports System.IO لابد من كتابه هذا الكود فى منطقه التصريحات العامه اى فوق الموديول قبل تطبيق اى مثال لتقوم باستيراد الدوال
Module module1

    Sub main()

        Dim W As New DirectoryInfo("C:\\")
        Dim D() As DirectoryInfo = W.GetDirectories
        Dim i As Integer
 For i = 1 To D.Length - 1

   'يقوم الكود التالى بجلب مسار كل المجلدات الموجوده فى البارتشنC
  Console.WriteLine("The Path Files into [C] :{0}", D(i).FullName)

   'يقوم الكود التالى بجلب اسماء كل المجلدات الموجوده فى البارتشنC
  Console.WriteLine("The Files Name into [C] :{0}", D(i).Name)

   'يقوم الكود التالى بجلب وقت وتاريخ اخر مره تم الدخول فيها على المجلد
  Console.WriteLine("The Last Use Time Files into [C] :{0}", D(i).LastAccessTime)

   'يقوم الكود التالى بجلب وقت اخر مره تم فيها التعديل على ملفات داخل الـC من حيث انشاء ملف بداخله وحذه ملف ..الخ
  Console.WriteLine("The Last Edit Time into Files:{0}", D(i).LastWriteTime)

   'لجلب وقت انشاء المجلد
  Console.WriteLine("The Creation Time Files into [C] :{0}", D(i).CreationTime)

   'لعمل خط يفصل بين معلومات كل مجلد على شاشه الكونسول
  Console.WriteLine("-------------------------------------------------")

Next
Console.ReadKey()
    End Sub
End Module
```

قمنا فى المثال السابق باستدعاء الدوال فى منطقه التصريحات العامه عن طريق الكود **Imports System.IO**
لابد من كتابه هذا الكود فى منطقه التصريحات العامه (اى فوق الموديول) قبل تطبيق اى مثال على هذه الدوال

ثم قمنا بحجز مخزن من النوع **DirectoryInfo** وامرناه ان يتعامل مع البارتشن **C:**
ثم قمنا بحجز مصفوفه ايضا من النوع **DirectoryInfo**
وذلك لتحمل معلومات المجلدات فى البارتشن **C:**
ثم قمنا بعمل دواره تقوم بعرض
معلومات المجلدات التى فى البارتشن **C:**
من حيث اسماء هذه المجلدات ومسار كل مجلد وتاريخ ووقت انشاءه وتاريخ ووقت التعديل عليه
ثم امرنا الكونسول بعمل خط يفصل بين معلومات كل ملف على شاشه الدوس

المثال الثانى:-

مثال يوضح كيفيه انشاء وحذف مجلد '

```
Imports System.IO

Module module1

    Sub main()


'طريقة الانشاء
        Dim W As New DirectoryInfo("C:\\Malek")
        If W.Exists = False Then   ' التاكد من كون الملف غير موجود اولا
            W.Create()  يفيد الكود بانشاء الملف عند عدم تحقق جمله الشرط (اى عدم وجود ملف بنفس الاسم)
            Console.WriteLine("Creation Done ")
            Console.ReadKey()
        Else


'طريقة الحذف
       If W.Exists = True Then   'جمله شرطيه تفيد بانه لو وجد ملف بنفس الاسم فى نفس المسار
                W.Delete()          '  يفيد الكود بحذف هذا الملف عند تحقق جمله الشرط
                Console.WriteLine("Deleted Done ")
                Console.ReadKey()
            End If
        End If


    End Sub
End Module
```

فى المثال السابق قمنا بحجز مخزن نوعه **DirectoryInfo** وامرناه ان يتعامل مع المسار **C:\\Malek**
ثم قمنا بعمل جمله شرطيه للتأكد من وجود مجلد فى البارتشن **C** بهذا الاسم **Malek** ام لا
تنص الجمله على انه عند عدم وجود هذا المجلد قم بانشاء
ثم قمنا بعمل جمله شرطيه اخرى تنص على انه ان وجد الملف قم بحذفه
لنجمع مابين كيفيه انشاء المجلدات وكيفيه حذفها فى مثال واحد.

المثال الثالث:-

مثال على نقل مجلد من مكان لاخر'

```
      Imports System.IO  '<< تذكير لك لاتنسى هذا الكود
Module module1

    Sub main()


        Dim W As New DirectoryInfo("C:\\Meko")
        If W.Exists = False Then
            W.Create()
            Console.WriteLine("Creation Done ")
            Console.ReadKey()

        Else

            If W.Exists = True Then
                W.MoveTo("c:\\windows\Meko")

                Console.WriteLine("Moved Done ")
                Console.ReadKey()
            End If
        End If
    End Sub
End Module
```

فى المثال السابق

قمنا اولا بانشاء مجلد بالاسم **Meko** فى البارتشن **C** كما فعلنا فى المثال الثانى

ثم قمنا بعمل جمله شرطيه تنص على انه عند وجود هذا المجلد قم بنقله الى المسار

C:\\windows\Meko

ومن الممكن ايضا ان نقوم بنقله مع تغيير اسمه عن طريق كتابه المسار الذى سيتحرك اليه هكذا

C:\\windows\اسم المجلد

المثال الرابع:-

مثال يوضح كيفيه التعامل مع الملفات الى بداخل مجلدات الـDirectoryInfo '

```
    Imports System.IO
Module module1

    Sub main()


        Dim W As New DirectoryInfo("C:\\Windows")
        Dim F() As FileInfo = W.GetFiles
        Dim I As Integer

 For I = 1 To F.Length - 1
    Console.WriteLine("The Name :{0}", F(I).Name) ' لمعرفه اسم الملف
    Console.WriteLine("The Path :{0}", F(I).FullName) ' لمعرفهمسار الملف
    Console.WriteLine("The Extention :{0}", F(I).Extension) ' لمعرفه امتداد الملف
    Console.WriteLine("The Attributes :{0}", F(I).Attributes) ' لمعرفهحاله الملف مخفى ام لا
    Console.WriteLine("----------------------------------------------------")
  Next
        Console.ReadKey()



    End Sub
End Module
```

قمنا بحجز مخزن من النوع DirectoryInfo ثم امرناه التعامل مع المسار التالى C:\Windows
ثم قمنا بحجز مصفوفه من النوع FileInfo لتحمل مواصفات الملفات التى بداخل المجلد Windows
ثم قمنا بحجز مخزن عددى لانشاء دواره به
تقوم هذه الدواره بعرض خصائص الملفات التى بداخل المجلد كما شرحنا فى الكود اعلاه
ثم امرنا الكونسول بوضع خط (-----) يفصل بين معلومات كل ملف من الملفات الموجوده فى المجلد Windows

المثال الخامس :-

مثال يوضح كيفيه عمل تشفير لملف معين عن باقى مستخدمين الويندوز'

```
Imports System.IO
Module module1

    Sub main()


'طريقه تشفير الملف
Console.Write("Enter The File Path U like To Encrypt: ")
        Dim Path As String = Console.ReadLine
        Dim F As New FileInfo(Path)
        If F.Exists = True Then
            F.Encrypt() 'هذا هو كود التشفير
            Console.WriteLine("Done!")
            Console.ReadKey()
        Else
         If F.Exists = False Then Console.WriteLine("The File Not Exists")
         Console.ReadKey()

     End If
```

فى المثال السابق
امرنا الكونسول ان يسأل المستخدم عن كتابه مسار الملف الذى يريد تشفيره
ثم قمنا بحجز مخزن نصى وامرناه ان يحمل قيمه ما ادخله المستخدم (المسار)
ثم قمنا بانشاء مخزن من النوع **FileInfo** ويتعامل مع المسار الذى ادخله المستخدم
ثم قمنا بعمل جمله شرطيه تنص على ان كان الملف موجود فى المسار صحيحا كما ادخله المستخدم قم بتشفيره
ثم انشأنا جمله شرطيه اخرى تنص على انه عند عدم وجود الملف فى المسار المحدد اظهر رساله تفيد
بان الملف غير موجود فى المسار المحدد
ثم نأتى الى طريقه فك التشفير
سيكون نفس كود التشفير ولكن نضع **F.Decrypt()** بدلا من **F.Encrypt**
تابع معى .....

'طريقه فك التشفير عن الملف

```
  Console.Write("Enter The File Path U like To Encrypt: ")
        Dim Path As String = Console.ReadLine
        Dim F As New FileInfo(Path)
        If F.Exists = True Then
            F.Decrypt() 'هذا هو كود فك التشفير عن الملف
            Console.WriteLine("Done!")
            Console.ReadKey()
        Else
            If F.Exists = False Then Console.WriteLine("The File Not Exists")
            Console.ReadKey()
        End If
    End Sub
End Module
```

المثال السادس :-

'مثال يقوم بعرض مجلدات وملفات مابداخل هذه المجلدات فى مسار معين

```
     Imports System.IO

Module module1

    Sub main()

        Dim WinDire As New DirectoryInfo("C:\Windows")
        Dim Drs() As DirectoryInfo = WinDire.GetDirectories

        Dim F() As FileInfo
        Dim B As Integer
        Dim I As Integer
        On Error Resume Next ' تم وضعها لعدم وقوف البرنامج اذا تواجد ملف محمى من الويندوز
        For I = 1 To Drs.Length - 1
            Console.WriteLine(Drs(I).Name)
            Console.WriteLine("----------------------------")
            F = Drs(I).GetFiles
            For B = 1 To F.Length - 1
                Console.WriteLine(F(B).Name)

            Next
        Next
        Console.ReadKey()


    End Sub
End Module
```

ركز كويس الكود ده غريب شويه عليك
قمنا بحجز مخزن من النوع DirectoryInfo وامرناه التعامل مع المسار C:\Windows
ثم قمنا بعمل مصفوفه من النوع DirectoryInfo تقوم بحمل المجلدات التى بداخل المسار
ثم قمنا بحجز مصفوفه اخرى من النوع FileInfo لتحمل بيانات الملفات التى بداخل كل مجلد من المجلدات
ثم قمنا بعمل فكره بسيطه
وهى عمل دوارتين بداخل بعض واحده تعرض المجلدات والاخرى تعرض مابداخل كل مجلد منهم
ولكن اذا وجد ملف محمى من الويندوز سوف يقوم بايقاف البرنامج واظهار خطأ
وللتغلب على هذه المشكله لابد من وضع الكود
On Error Resume Next
اى عند وجود خطأ تجاهله واذهب الى مابعده

المثال السابع :-

سوف نتناول فى هذا المثال بعض الدوال الاخرى التى تتعامل مع الملفات والمجلدات
وهى دوال منفصله عن دوال الـ InputOutput
اذا لانحتاج لكتابه الكود Imports System.IO
من هذه الدوال

الداله MkDir:
تقوم هذه الداله بانشاء مجلدات فى مسار معين

الداله Rename:
تستخدم هذه الداله فى تغير اسم المجلدات

الداله RmDir:
تستخدم لمسح مجلد معين

الداله Kill:
تستخدم هذه الداله لمسح ملف وليس مجلد ركز كويس

الداله CurDir:
تستخدم لعرض مسار البرنامج

تابع معى المثال الاتى لتعرف كيفيه استخدام كل داله من هذه الدوال

مثال يحتوى على دوال اخرى تتعامل الملفات والمجلدات '

```
Module module1

    Sub main()

        MkDir("c:\\Meko")                   ' لانشاء مجلد
        Rename("C:\\Meko", "C:\\Malek")   'لتغيي اسم مجلد معين
        RmDir("C:\\Malek")              ' لمسح مجلد معين
        Kill("c:\\dsn.txt")             '   لمسح ملف معين
        Dim A As String = CurDir()  'يفيد الكود الحصول على مسار المجلد المحتوى على البرنامج الذى قمنا بتصميمه
        Console.WriteLine(A)
        Console.ReadKey()

    End Sub
End Module
```

المثال الثامن:-

STREAM WRITER

مثال يوضح كيفيه الكتابه بداخل ملفات التكست من خلال الـ StreamWriter'

```
Imports System.IO
Module module1

    Sub main()
      Dim Sw As StreamWriter = File.AppendText("C:\\Agc.txt")  'لانشاء ملف يقبل الاضافه كل مره

        Console.WriteLine("Enter UR Name: ")
        Dim M As String = Console.ReadLine
        Sw.WriteLine(M)  ' للكتابه داخل الملف بداخل سطر مستقل
        Sw.WriteLine("---------")
        Sw.Close()  'لعمل تحديث على البيانات داخل الملف واغلاقه

    End Sub
End Module
```

فى المثال السابق

قمنا باستدعاء دوال **IO** فى منطقه التصريحات العامه

Imports System.IO

ثم قمنا باستعمال الداله **StreamWriter** تحت الصب الرئيسى

عن طريق حجز مخزن نسخه منها عن طريق الكود

Dim Sw As StreamWriter

وجعلناها تنشأ ملف تكست على المسار C:\\Agc.txt
وهذا الملف يكون قابل للاضافه كل مره (اى عند تشغيل البرنامج كل مره يضيف على البيانات القديمه لهذا الملف)
ومن الممكن عمل ملف قابل للاضافه كل مره بطريقه اخرى

```
Dim Sw As New StreamWriter("C:\\Agc.txt", True)
```

والمعنى هنا اى قم بحجز مخزن من النوع **StreamWriter**
واجعل قيمه الاضافه بداخله كل مره **True**

ومن الممكن عمل ملف غير قابل للاضافه (اى كل مره يحذفه وينشأ ملف جديد) وذلك عن طريق الكود التالى

```
Dim Sw As New StreamWriter("C:\\Agc.txt", False)
```

والمعنى هنا اى قم بحجز مخزن من النوع **StreamWriter**
واجعل قيمه الاضافه بداخله كل مره **False**
هكذا يصبح الملف غير قابل للاضافه ففى كل مره يعمل فيها البرنامج يقوم بحذفه وانشاء ملف غيره بنفس الاسم

ثم امرنا الكونسول ان يطلب من المستخدم اسمه عن طريق الكود

```
Console.WriteLine("Enter UR Name: ")
```

ثم قمنا بحجز مخزن نصى ليحمل القيمه المدخله من المستخدم عن طريق الكود

```
Dim M As String = Console.ReadLine
```

ثم امرنا المخزن **SW**
بكتابه ما بقيمه المخزن النصى **M**على سطر مستقل فى ملف التكست
ثم بعمل خط **(----)** فى سطر مستقل فى ملف التكست ليفصل بين الكلام عن طريق الكود

```
Sw.WriteLine(M)
Sw.WriteLine("---------")
```

ثم امرنا المخزن **Sw** ان يقوم بعمل تحديث على الملف ويغلقه عن طريق الكود

```
Sw.Close
```

ومن الممكن عمل تحديث على البيانات داخل ملف التكست بدون اغلاقه وذلك عن طريق كتابه الكود التالى

```
Sw.Flush()
```

الأن عرفنا كيف نقوم بالكتابه داخل ملفات التكست
يتبقى لنا كيفيه عرض مابداخل ملفات التكست
تابع معى... فى المثال القادم سوف اشرح لك
كيفيه قراءه مابداخل ملفات التكست وعرض هذه البيانات على شاشه الكونسو

المثال التاسع:-

STREAM READER

مثال يوضح كيفيه الكتابه بداخل ملفات التكست من خلال الـ StreamReader'

```
Imports System.IO

Module Module1
    Sub MAIN()

       Dim Sr As StreamReader = File.OpenText("C:\\Agc.txt")  'لفتح ملف التكست
        Dim H As String = Sr.ReadLine ' لقراءه سطر واحد من الملف
        Console.WriteLine(H)
        Console.ReadKey()

    End Sub
End Module
```

فى المثال السابق

قمنا بحجز مخزن من النوع StreamReader

وامرنا بفتح ملف التكست (الذى تم انشائه فى المثال الثامن) عن طريق الكود

`Dim Sr As StreamReader = File.OpenText("C:\\Agc.txt")`

ومن الممكن فتح الملف بطريقه اخرى عن طريق الكود

`Dim Sr As New StreamReader("C:\\sgc.txt")`

ثم قمنا بحجز مخزن نصى وجعلنا المخزن القارىء من الملفات ان يضع به اول سطر من الملف عن طريق الكود

`Dim H As String = Sr.ReadLine`

فان الداله **ReadLine**

تقوم بقراءه اول سطر من الملف

ومن الممكن ايضا ان نقرا كل مافى الملف بهذه الداله ولكن عن طريق دواره

تابع معى فكره الدواره.......

```
Dim C As String
Do While Not Sr.EndOfStream
C = Sr.ReadLine
Console.WriteLine(C)
Loop
```

تقوم هذه الدواره بقراءه سطر واحد فى كل لفه حتى تصل الى نهايه الملف

واذا اردنا ان نقرأ كل مافى الملف بدون دواره يكون عن طريق الكود التالى
Dim H As String = Sr.ReadToEnd

فان الداله **ReadToEnd**
تقوم بقراءه كافه البيانات فى الملف حتى نهايته

ومن الممكن ايضا ان نحصل على ارقام الاسكى لكل حرف فى الكلمات المكتوبه داخل الملف عن طريق الكود
Dim B As String = Sr.Read

فان الداله **Read**
تقوم بجلب ارقام الاسكى لكل حرف من حروف الكلمات المكتوبه داخل الملف
اذا اردت ان تعرف ماهى ارقام الاسكى راجع **(الوحده الخامسه)** فى تشفير الرسائل

المثال العاشر:-

FILE STREAM

تدمير الملفات:-
قبل ان نبدأ فى شرح ذلك الجزء لابد ان تعلم
ان اى ملف فى العالم يتعامل من الحاسب الالى فهو يتكون من الارقام(1وصفر)
وتكون هذه الارقام مرتبه بطريقه معينه فى كل ملف
انظر الصوره التاليه

ترتيبه الـ Byte لملف معين

1	0	1	0
0	1	0	1
1	0	1	1
0	1	0	0

فهذه عباره عن صوره توضيحيه لترتيب البايت لملف معين
فاذا اختلفت الترتيبه لهذا الملف اصبح الملف معطوب ولايصلح للعمل
فاذا قمنا باعاده ترتيبها كما كانت عاد الملف الى طبيعته
وهذه طريقه من طريق التشفير للملفات
من الممكن ان نستخدمها فى برامجنا ولا نستغنى عنها فمن الممكن ان نصمم برنامج
متصل بقاعده بيانات معينه ونجعله عند الاغلاق يقوم بتدمير قاعده البيانات لكى لا يطلع عليها احد
ثم عند تشغيل البرنامج يقوم باعاده ترتيب البايت الخاص بها كما كان لتعمل بطبيعتها
فهذه فكره من افكار الحمايه .... وتختلف الافكار كلا على حسب طريقته

مارأيك اذا قمنا بتغيير ذلك الترتيبه؟
لصوره مثلا ونرى النتيجه سويا !!!
تابع معى .....

' **مثال يوضح كيفيه تدمير الملفات من خلال الــ FileStream**

```
Imports System.IO
Module Module1

    Sub main()
        Dim FS As New FileStream("c:\\Malek.jpg", FileMode.Open)
        Dim H(2) As Byte
        Dim F(2) As Byte
        FS.Read(H, 0, 2)
        FS.Seek(-2, SeekOrigin.End)
        FS.Read(F, 0, 2)

        FS.Seek(-2, SeekOrigin.End)
        FS.Write(H, 0, 2)

        FS.Seek(0, SeekOrigin.Begin)
        FS.Write(F, 0, 2)
        FS.Close()
        Console.WriteLine("Done")
        Console.ReadKey()
    End Sub

End Module
```

فى المثال السابق

قمنا بحجز مخزن من النوع فايل ستريم **FileStream**

وامرناه ان يقوم بفتح الصوره بالفايل مود ليتعامل مع البايت لها(التى على المسار التالى) C:\\Malek.jpg

وذلك عن طريق الكود

Dim FS As New FileStream("c:\\Malek.jpg", FileMode.Open)

ثم قمنا بانشاء مصفوفتين من النوع بايت (الذى يحمل البيانات التى نوعها بايت)

وكلا من المصفوفتين يحتوى على جزئين وذلك لتحمل كل مصفوفه منهم حزئين من ترتيبه البايت للصوره او من الممكن ان نزيد من اجزائها وذلك حسبما يريد المصمم.

ثم امرنا مخزن الفايل استريم ان يضع اول جزئي من الترتيبه فى المصفوفه **H**

FS.Read(H, 0, 2)

ثم امرناه ان يتحرك الى اخر الترتيبه ويرجع خطوتين

FS.Seek(-2, SeekOrigin.End)

ثم امرناه ان يضع اخر اتنين بايت من الترتيبه فى المصفوفه **F**

FS.Read(F, 0, 2)

ركز!!!!!

اصبح الان المصفوفه **H** تحمل اول **2** بايت فى الترتيبه

والمصفوفه **F** تحمل اخر **2** بايت فى الترتيبه

والان لكى نقوم بتدمير الصوره علينا تبديل مافى كل مصفوفه مكان الاخرى (اى تبديل اول اتنين بايت مع اخر اتنين بايت).

تابع معى..... ولاتسهى !!!!

ثم امرنا المخزن فايل ستريم بالذهاب الى اخر الترتيبه والرجوع خطوتين

ثم كتابه ماتحمله المصفوفه **H**

```
FS.Seek(-2, SeekOrigin.End)
FS.Write(H, 0, 2)
```

ثم امرناه بالذهاب لبدايته الترتيبه وان يضع بها ماتحمله المصفوفه F وان يغلق الملف

```
FS.Seek(0, SeekOrigin.Begin)
FS.Write(F, 0, 2)
FS.Close()
```

ثم امرنا الكونسول ان يظهر رساله بانه تم التدمير

```
Console.WriteLine("Done")
```

انظر الى النتيجه حين اقوم بفتح الصوره



هكذا قد قمنا بتدمير الصوره
فانك عند تشغيل المثال مره اخرى سوف يقوم بعكس الترتيبه مره اخرى
ليعود كل بايت الى مكانه الصحيح فتعمل الصوره

انظر الى النتيجه حين قمت بتشغيل البرنامج مره اخرى



والسؤال هنا
لماذا عادت الصوره لطبيعتها مره اخرى ؟
وكيف اجعلها لا تعود لطبيعتها الا بأمر معين منى ؟

الاجابه !!!
اولا : عادت الصوره لطبيعتها عند تشغيل البرنامج مره اخرى وذلك لان الاكواد قامت بعكس ماقامت به اول مره
فعاد كل بايت الى طبيعته فى ترتيبه الصوره وذلك يرجع لانها تتحرك بمقدار ثابت داخل الترتيبه وهو 2
لان كل مصفوفه تحتوى على جزئين وتقوم بحمل 2 بايت ووضعهم فى مكان ثابت
وهذه هى فكره تدمير قاعده البيانات لبرنامج معين بعد اغلاقه واعادتا لطبيعتها عند تشغيل البرنامج
ليصبح هو البرنامج الوحيد الذى يمكن عن طريقه قراءه مافى قاعده البيانات كما قولنا سابقا !!

ثانيا : لكى تجعل الملف لايعود لطبيعته مع تشغيل البرنامج مره اخرى
تجد الاجابه عليه فى المثال القادم **تابع معى ....**

المثال الحادى عشر:-

فى هذا المثال سوف نتعمق اكثر فى كيفيه تدمير الملفات
سوف نجعله يقوم بتدمير الملفات عن طريق رقم معين ويعيد الملف الى اصله بنفس الرقم
وذلك عن طريق جعل المستخدم يختار اجزاء المصفوفه
تابع معى... لتصل اليك الفكره اكثر

مثال يوضح كيفيه تدمير الملفات برقم معين واعادتها لطبيعتها بنفس الرقم'

```
Imports System.IO
Module Module1

    Function Encryption(ByVal Fn As String, ByVal L As Integer) As String

        If L > 20 Then
            Return "Choose Another No"
        End If

        If File.Exists(Fn) = False Then
            Return "File Not Found"
        End If

        Dim fS As New FileStream(Fn, FileMode.Open)
        Dim H(L) As Byte
        Dim F(L) As Byte
        fS.Read(H, 0, L)
        fS.Seek(-L, SeekOrigin.End)
        fS.Read(F, 0, L)
        fS.Seek(-L, SeekOrigin.End)
        fS.Write(H, 0, L)
        fS.Seek(0, SeekOrigin.Begin)
        fS.Write(F, 0, L)
        fS.Close()
        Return "Done"
        Console.ReadKey()

    End Function

    Sub main()
        Console.WriteLine("Enter File Path")
        Dim A As String = Console.ReadLine

        Console.WriteLine("Enter Encryption No")
        Dim B As Integer = Console.ReadLine

        Dim S As String = Encryption(A, B)
        Console.WriteLine(S)
        Console.ReadKey()

    End Sub

End Module
```

فى المثال السابق
قمنا بانشاء فانكشن يحتوى على **2** معامل
المعامل الاول `Fn As String` من النوع النصى ليستقبل مسار الملف
والمعامل `L As Integer` من النوع العددى ليحمل قيمه اجزاء المصفوفتين
التى سوف نتنقل بها بين ترتيبه البايت كما شرحنا فى المثال السابق
يعود هذا الفانكشن بقيمه نصيه. تم انشاءه عن طريق الكود

```
Function Encryption(ByVal Fn As String, ByVal L As Integer) As String
```

ثم وضعنا بداخله عده شروط لتلاشى الاخطاء
الشرط الاول لو كان قيمه المعامل **L** اكبر من 20
عد برساله للمستخدم تفيد بان يختار رقم اقل من ذلك

```
If L > 20 Then
Return "Choose Another No"
```

وفائده هذا الكود انه يجبر المستخدم بعدم الزياده فى اجزاء المصفوفهلكى لا تكون اكبر من عدد خلايا ترتيبه البايت للملف فيحدث خطأ ويتوقف البرنامج عن العمل.

ثم وضعنا شرط اخر
لو كان الرقم اقل من 20 والملف غير موجود بالمسار الذى وضعه المستخدم فى المعامل **Fn**
قم باظهار رساله تفيد بان الملف غير موجود

```
End If
If File.Exists(Fn) = False Then
Return "File Not Found"
```

واذا لم تحدث هذه الاخطاء وكان المسار والرقم صحيح
اذا قم بالتنقل بين ترتيبه البايت للملف وتغيير هذه الترتيبه
على حسب الرقم المدخل لك والمحمل فى المعامل **L** كما حدث فى المثال السابق

```
End If
Dim fS As New FileStream(Fn, FileMode.Open)
Dim H(L) As Byte
Dim F(L) As Byte
fS.Read(H, 0, L)
fS.Seek(-L, SeekOrigin.End)
fS.Read(F, 0, L)
fS.Seek(-L, SeekOrigin.End)
fS.Write(H, 0, L)
fS.Seek(0, SeekOrigin.Begin)
fS.Write(F, 0, L)
fS.Close()
Return "Done"
Console.ReadKey()
```

لاحظ هذا الكود السابق !!!
هو نفس كود التدمير فى المثال السابق ولكن يختلف عنه
فى اننا وضعنا مافى المعامل **L**
بدلا من وضع الرقم الثابت 2

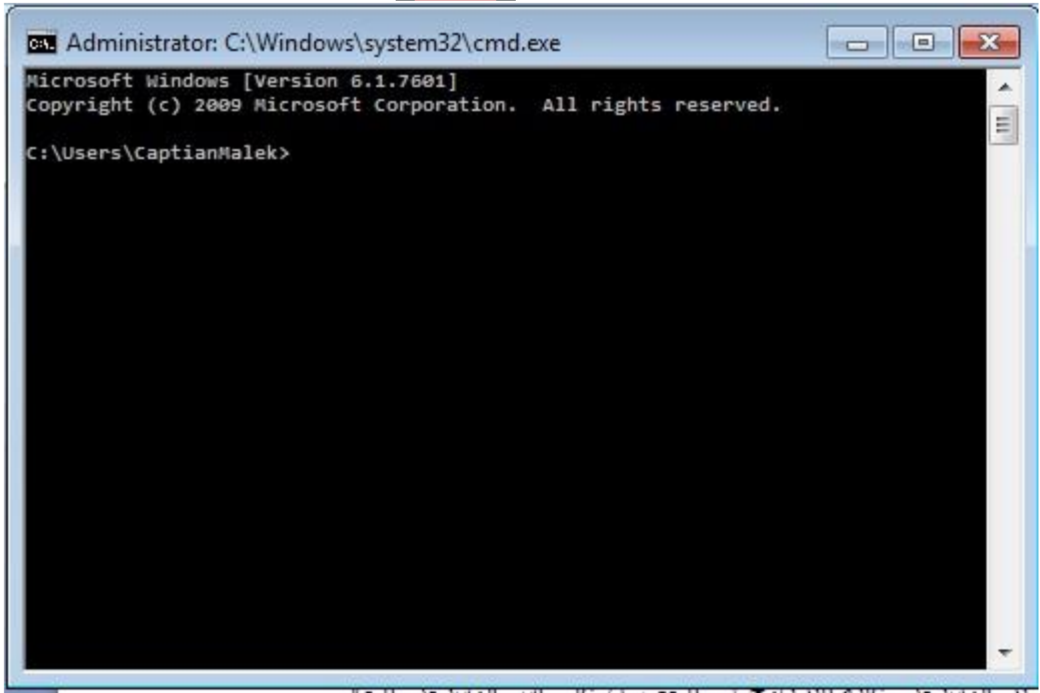
انتهت الوحده التاسعه

الوحده العاشره :-

ENVIRONMENT

تهدف هذه الوحده الى :-
كيفيه التعامل مع بيئه البرنامج من حيث مثلا ... الحصول على مسار البرنامج او نقل ملفات البرنامج من مكان لمكان اخر او الحصول على اسم جهاز الكمبيوتر المستخدم عليه البرنامج او الحصول على اسم المستخدم الحالى لجهاز الكمبيوتر او الحصول على عدد البروسيسورات المستخدمه فى الجهاز او الحصول على حاله برنامج ان كان اوف لاين او اون لاين .... الخ
او عمل حمايه للبرنامج بحيث لايعمل الا بعد اعطاءه امر معين من الـ CMD
و هى تعتبر شاشه الدوس الخاصه بالويندوز

الصوره التاليه توضح شاشه الدوس الخاصه بالويندوز (CMD)



ويتم الحصول على هذه الشاشه من خلال الدخول الى المسار الطبيعي لها و هو
C:\Windows\System32
او بالطريقه الاسهل من قائمه استارت
Start >> Run >> نكتب الامر Cmd

اذا تعالى بنا فى المثال الاول نتعلم كيفيه حمايه البرنامج عن طريق عدم تشغيله الا بعد اعطاءه امر من شاشه الدوس الخاصه بالويندوز

المثال الأول :-

يتحدث هذا المثال عن كيفيه عمل حمايه للبرنامج بطريقه معينه
و هى عدم تشغيل البرنامج الا بعد اعطاءه امر معين من شاشه الدوس الخاصه بالويندوز وان لم يتم اعطاءه هذا الامر يظهر البرنامج خطأ معين
لكى يقوم بتشتيت فكر الكراكرز الذين يقومون بفك الحمايه عن البرامج فعند ظهور الخطأ ف البرنامج يظن ان البرنامج به اخطاء وغير صالح للعمل .....
ويتم عمل ذلك الحمايه عن طريق داله من دوال بيئه البرنامج وتسمى هذه الداله **CommandLine**
اخى القارىء من فضلك اترك كل تركيزك مع هذا المثال لانه من اهم الامثله التى وردت فى هذا الجزء من الكتاب و هى فكره جديده لحمايه البرامج من الكراكرز

مثال بسيط يوضح كيفيه عمل حمايه بالكوماند لاين'

```
Dim X As String = Environment.CommandLine
        If InStr(X, "Agc") > 0 Then

            Console.WriteLine("Welcome to Agc Academy")
            Console.ReadKey()
            End
        End If
        Err.Raise(9)
```

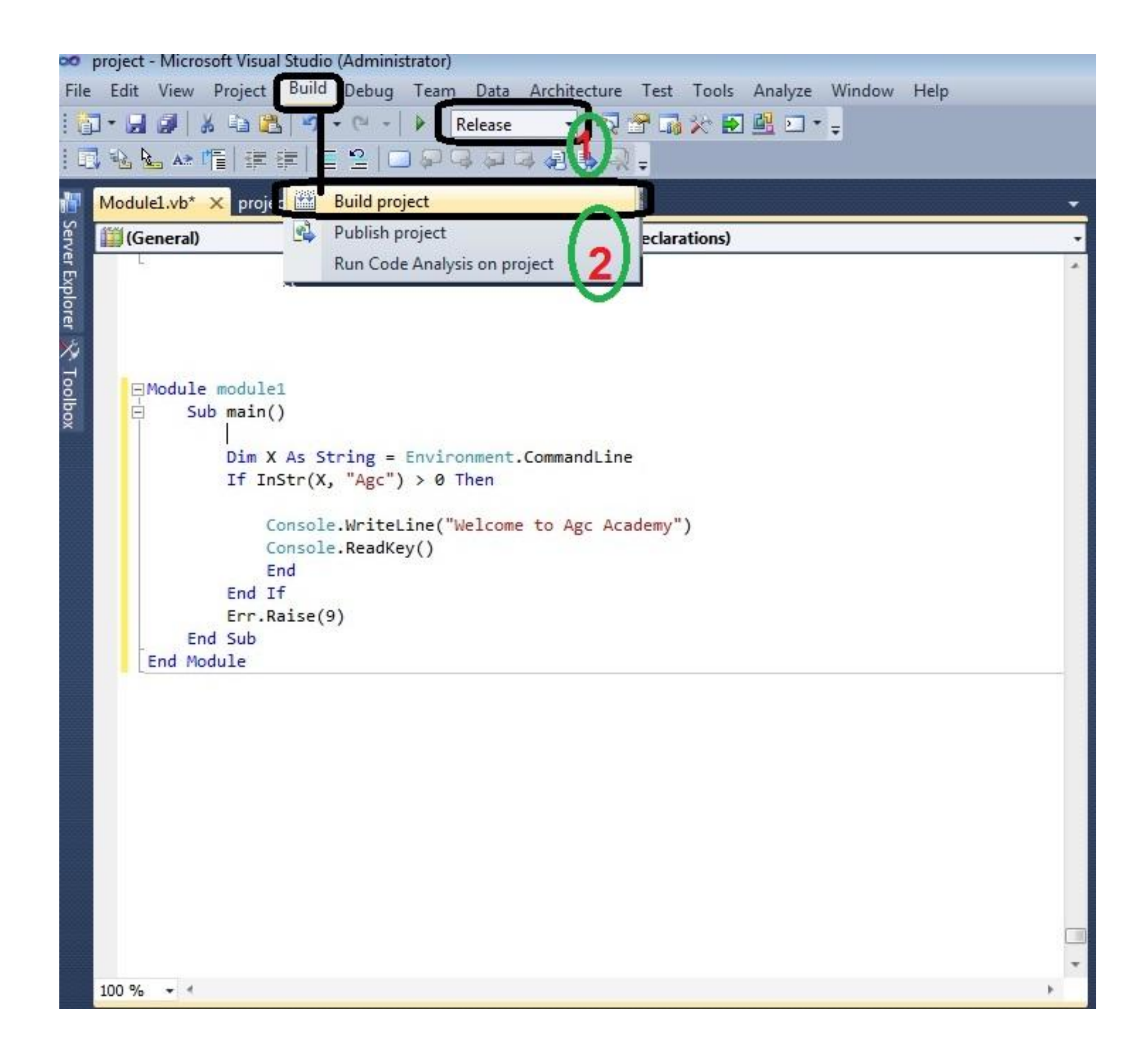
فى المثال السابق
قمنا بحجز مخزن نصى وجعلناه مساويا للكوماند لاين فى بيئه البرنامج **(Environment.CommandLine)**
معنى ذلك اننا دخلنا فى محتويات البرنامج نفسه انظر الى ماحدث بعد ذلك
قمنا بانشاء جمله شرطيه تنص على انه اذا وجد فى هذا المخزن النصى كلمه **Agc**
قم بكتابه رساله تفيد بأنه تم الدخول الى البرنامج بنجاح
وان كانت لا تحتوى على كلمه **Agc**
قم بتفجير الخطأ 9
اذا كيف ندخل الى محتويات البرنامج ؟ ونجعله يحتوى على هذه الكلمه ليعمل ؟
الاجابه هى الدخول الى شاشه الدوس واعطاء الامر بطريقه معينه

قف هنا !!

تعالى بنا نطبق المثال بطريقه عمليه
اولا : قم بكتابه هذه الاكواد جيدا وافهم معنى كلا منهم

ثانيا : قم باستخلاص البرنامج وتحويله **EXE**
وذلك عن طريق تحويله اولا الى Relase
ثم من القائمه الرئيسيه لبرنامج الفيجوال
Build >> Build Project
الان البرنامج تم استخلاصه اذهب اليه فى مجلد المشروع سوف تجده فى المجلد
Relase

اليك بعض الصور التوضيحيه بالخطوات لكيفيه تحويل البرنامج الى EXE



هكذا قم تم تحويل البرنامج

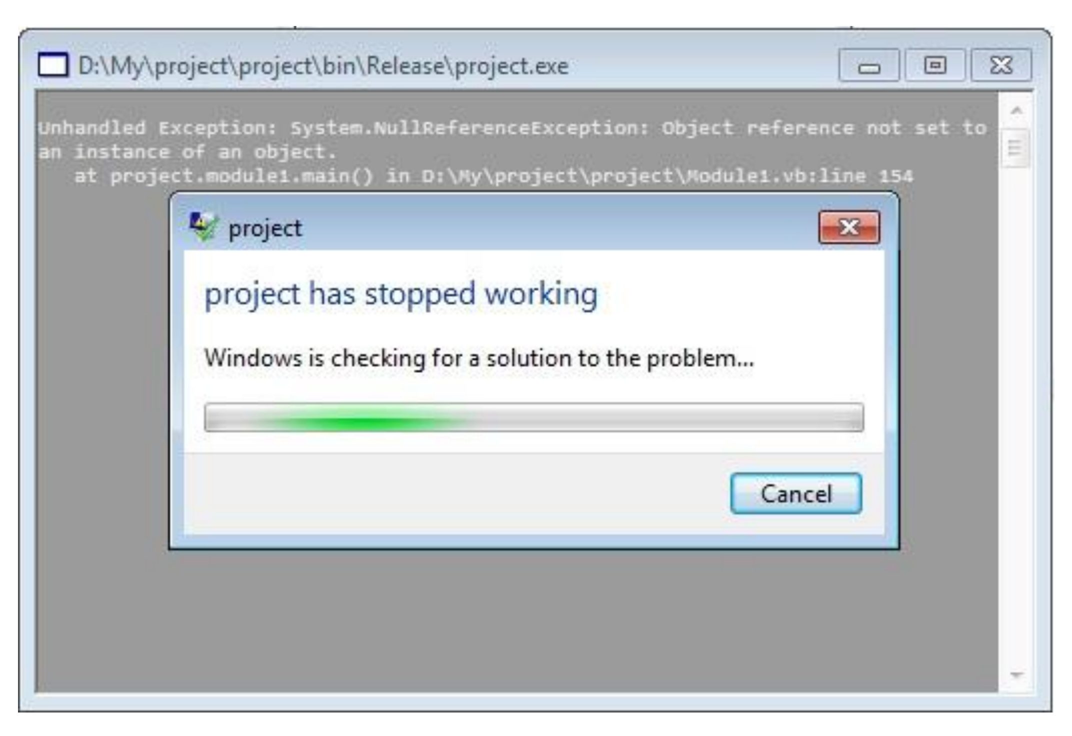
الان اذهب الى مجلد المشروع فى المجلد **Bin**

ستجد البرنامج بأسمه >> Relase >> Bin

قم بتشغيله الان !!! سوف يعطيك خطأ كأن البرنامج معطوب ولا يعمل وذلك لاننا لم نعطيه القيمه

Agc

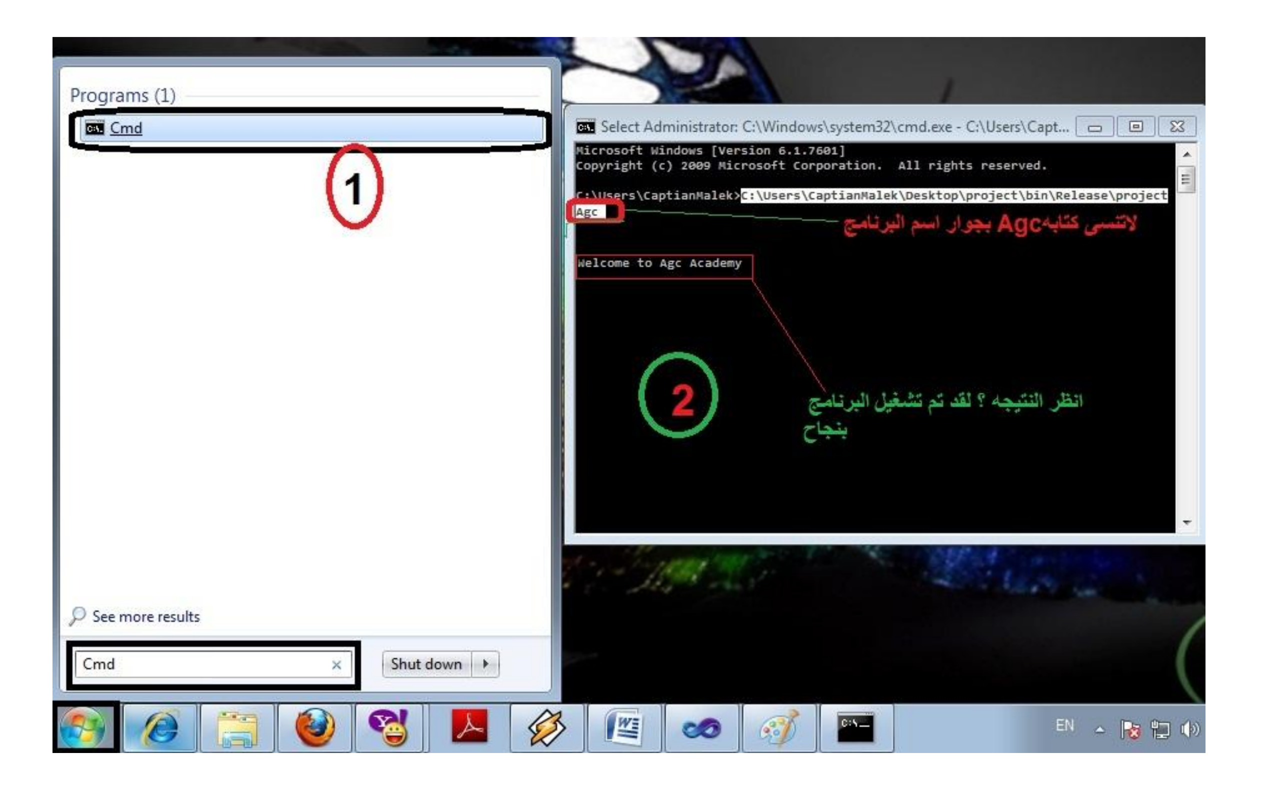
انظر الصوره التاليه :-



لابد ان نعطيه القيمه **Agc**

فى شاشه الدوس الخاصه بالويندوز لكى يعمل بشكل طبيعي

انظر الصوره التاليه



فى الخطوه الاولى :

قمنا بتشغيل شاشه الدوس من قائمه ستارت كما شرحنا مسبقا

فى الخطوه الثانيه :

قمنا بكتابه مسار البرنامج فى شاشه الدوس كما ترى فى الصوره
ثم قمنا بكتابه اسم البرنامج والكلمه السريه بجوار اسمه
فكما ترى اسم البرنامج عندى **Project** وقمت بكتابه **Agc** بجواره
وكانت النتيجه كما ترى
Welcom to Agc Academy
اى تم عمل البرنامج بنجاح

هذه كانت طريقه الحمايه للبرنامج
اخى القارىء اتمنى ان تكون قد قمت بتطبيق هذاالمثال جيدا وفهمت كل مافيه من اكواد وخطوات والان دعنا ننتقل الى توضيح باقى عمل دوال بيئه البرامج فى الامثله القادمه

المثال الثانى :-

الداله CurrentDirectory :
تستخدم فى عرض مسار المشروع ونقل المشروع من مكان الى اخر
تابع معى المثال لتعرف كيفيه استخدامها

مثال يوضح كيفيه نقل المشروع من مكان لاخر'

```
'لعرض المسار الحالى للبرنامج

Dim M As String = Environment.CurrentDirectory
Console.WriteLine(M)


'لنقل البرنامج من مساره الاصلى الى البارتشن C

Environment.CurrentDirectory = "C:\\"
Dim x As String = Environment.CurrentDirectory
Console.WriteLine(x)
Console.WriteLine("The Project File Moved Done")
Console.ReadKey()
```

فى المثال السابق
قمنا بحجز مخزن نصى وجعلناه مساويا بالداله التى تحمل مسار البرنامج
اصبح المخزن النصى الان يحمل مسار البرنامج
ثم امرنا الكونسول ان يعرض مافى المخزن على شاشه الدوس ليقوم بعرض المسار الحالى للبرنامج

ثم امرنا الداله **Environment.CurrentDirectory**
ان تحمل مسار البرنامج على البارتشن **C**
بذلك قمنا بتغيير مسار البرنامج بنجاح
ثم قمنا بحجز مخزن من النوع النصى
وجعلناه مساوى بالداله التى تحمل مسار البرنامج بعد التغيير
ثم امرنا الكونسول بعرض ما ف المخزن النصى ليعرض مسار البرنامج بعد التغيير

المثال الثالث:-

يضم معظم التعاملات مع بيئه البرنامج'

```
Dim A As String = Environment.MachineName 'للحصول على اسم الجهاز
Console.WriteLine(A)
Console.WriteLine("======================")

Dim B As Integer = Environment.ProcessorCount  'للحصول عدد البروسيسورات فى الجهاز
Console.WriteLine(B)
Console.WriteLine("======================")

Dim C = Environment.Version  'للحصول على رقم الاصدار
Console.WriteLine(C)
Console.WriteLine("======================")


Dim D = Environment.UserName  'للحصول على اسم المستخدم الحالى للويندوز
Console.WriteLine(D)
Console.WriteLine("======================")

Dim E As String = Environment.HasShutdownStarted  'لمعرفه حاله البرنامج يعمل ام مغلق
Console.WriteLine(E)
Console.WriteLine("======================")

Dim G = Environment.CurrentDirectory    'للحصول على مسار البرنامج الحالى تم شرحه مسبقا
Console.WriteLine(G)
Console.WriteLine("======================")


Dim H = Environment.WorkingSet     ' للحصول على حجم الميمورى للجهاز
Console.WriteLine(H)
Console.WriteLine("======================")
Console.ReadKey()
```

ملحوظه !!
لابد ان تقوم بتطبيق كلا من الاكواد السابقه جيدا
لتعرف ماهميه كل كود منهم نظريا
لان القراءه وحدها لاتكفى التطبيق يزيد من تركيز المعلومه فى عقلك

انتهت الوحده العاشره

الوحده الحاديه عشر:-

STRUCTURE

تهدف هذه الوحده الى :

تعليم كيفيه هيكله المتغيرات والفانكشن و كل الاكواد المعروفه فى الفيجوال
وكيفيه الاستفاده من هذه الهيكله وفائدتها فى البرمجه .

- يتم انشاء الاستراكشر او الهيكل فوق الصب الرئيسى او تحته
ولكن لابد ان يكون داخل الموديول ولك الاختيار فى انشاءه فوق او تحت الصب الرئيسى
تابع معى الامثله لتعرف ماهى اهميه الهيكله

المثال الاول:-

```
Module module1

    Structure Details
        Dim UserName As String
        Dim Password As String
        Dim Address As String
    End Structure

    Sub main()

        Dim D As Details
        Console.Write("Enter UserName:")
        D.UserName = Console.ReadLine()

        Console.Write("Enter Password: ")
        D.Password = Console.ReadLine


        Console.Write("Enter UR Address: ")
        D.Address = Console.ReadLine

        Console.WriteLine(D.UserName)
        Console.WriteLine(D.Password)
        Console.WriteLine(D.Address)
        Console.ReadKey()

    End Sub
End Module
```

فى المثال السابق

قمنا بانشاء هيكل بالاسم Details فوق الصب الرئيسى يحتوى على ثلاث مخازن من النوع النصى

ثم قمنا باستخدامه تحت الصب الرئيسى عن طريق الجمله

Dim D As Details

اى قم بحجز مخزن من نوع الاستراكشر او الهيكل

اصبح المخزن الان يحمل نسخه كامله من الهيكل بكل مخازنه

ثم امرنا الدوس ان يسأل المستخدم عن اسمه ثم يقوم بوضع القيمه المدخله فى اول مخزن فى الاستراكشر

ثم يسأل المستخدم عن قيمه الباسورد ويضعها فى المخزن الثانى من مخازن الاستراكشر

ثم يسأل عن رقم العنوان ويضعه فى المخزن الثالث من مخازن الاستراكشر

اصبحت مخازن الاستراكشر الان تحمل القيم المدخله من المستخدم (الاسم والباسورد والعنوان)

ثم امرنا الكونسول بعرض القيم التى بداخل كل مخزن من مخازن الاستراكشر

المثال الثانى:-

سوف نقوم فى هذا المثال بعمل برنامج عملى ملموس او بمعنى اصح مستعمل فى السوق

و هو عباره عن مثال يوضح اهميه الهيكله فى عمل برنامج يقوم بحساب النسبه المئويه والمجموع الكلى لطلبه فى مدرسه معينه **تابع معى ....**

مثال لعمل هيكل يقوم بحساب النسبه المئويه والمجموع الكلى لطلبه المدارس'

```
Module module1

    Structure Student

        Dim RealName As String
        Dim Math As Double
        Dim Physics As Double
        Dim Chemestry As Double
        Dim Piolgy As Double
        Dim Total As Double
        Dim Percent As Double

    End Structure


    Sub main()

        Dim S As Student
        Console.Write("Enter Ur Real Name : ")
        S.RealName = Console.ReadLine

        Console.Write("Enter Ur Math Degree : ")
        S.Math = Console.ReadLine

        Console.Write("Enter Ur  Physics : ")
        S.Physics = Console.ReadLine

        Console.Write("Enter Ur Chmestry Degree : ")
        S.Chemestry = Console.ReadLine
```

```
        Console.Write("Enter Ur Piolgy Degree : ")
        S.Piolgy = Console.ReadLine

        Dim T As Double
        T = S.Math + S.Physics + S.Chemestry + S.Piolgy
        S.Total = T
        Console.WriteLine("UR Total Degree : {0}", T)

        S.Percent = (T * 100) / 200
        Console.WriteLine("Ur Percent Degree : {0}%", S.Percent)
        Console.ReadKey()
    End Sub
End Module
```

فى المثال السابق

قمنا بعمل هيكل بالاسم **Student**

يحتوى هذا الهيكل على مخزن من النوع النصى لوضع الاسم به

وبه 6 مخازن اخرى من النوع العددى المزدوج الذى يقبل الارقام العشريه لادخال بهم قيم مجاميع المواد وحساب المجموع والنسبه المئويه لهذه المجاميع

ثم قمنا باستخدام هذا الهيكل تحت الصب الرئيسى

وذلك عن طريق حجز مخزن نسخه منه عن طريق الكود

Dim S As Student

اذا اصبح ذلك المخزن يحتوى على نسخه من الهيكل وكل مخازنه

ثم امرنا الكونسول بسؤال المستخدم عن ادخال درجه كل ماده من المواد

Math , Physics , Chmestry , Piolgy

ثم وضعنا كل قيمه من قيم المواد فى المخزن الموازى لها فى الهيكل

ليصبح الهيكل محملا بقيم مجاميع المواد السابقه والتى ادخلها المستخدم

ثم قمنا بوضع قيمه جمع مجاميع المواد فى مخزن عددى وجعلناه مساويا للمخزن المسئول عن المجموع الكلى للمواد فى الهيكل

اذا اصبح مخزن المجموع الكلى فى الهيكل يحتوى على مجموع كل المواد التى تم اداخلها من المستخدم

ثم الى المخزن الخاص بحساب النسبه المئويه لهذا المجموع

افترضنا ان كل ماده من المواد مجموعها النهائى هو 50

اذا 4 مواد * 50 = 200

يبقى الدرجه العظمى هى 200

وبما ان النسبه المئويه تساوى المجموع الاساسى مضروبا فى 100 مقسوما على الدرجه العظمى

لذلك قمنا بكتابه ذلك المعادله فى الكود

S.Percent = (T * 100) / 200

وتعنى هذه المعادله اى قم بضرب المجموع النهائى للطالب فى 100 ثم قم بقسمه الناتج على الدرجه العظمى 200 لنحصل على النسبه المئويه

ثم ضع ذلك النسبه المئويه فى المخزن الخاص بها فى الهيكل

ليصبح الهيكل الان

محملا بدرجه كل ماده من المواد و المجموع الكلى لهذه المواد والنسبه المئويه لهم

وفى النهايه امرنا شاشه الكونسول بعرض كل هذه القيم

ليحصل المستخدم على قيمه النسبه المئويه والمجموع الكلى له

اعرف ان المثال طويل وصعب لدرجه معينه ولكن اذا قمت بتطبيقه اكثر من مره سوف تفهمه تعالى بنا نطول المثال اكثر تابع معى ....

المثال الثالث:-

نفس المثال الثانى ولكن عن طريق استخدام الفانكشن فى الهيكله وحساب التقدير للمجموع

```
Module module1

    Structure Student

        Dim RealName As String
        Dim Math As Double
        Dim Physics As Double
        Dim Chemestry As Double
        Dim Piolgy As Double


        Function T() As Double
            T = Math + Physics + Chemestry + Piolgy
        End Function
        Function P() As Double
            P = (T() * 100) / 200

        End Function

        Function Rating() As String
            If P() < 50 Then Rating = "Drop"
            If P() >= 50 And P() < 60 Then Rating = "Passed"
            If P() >= 60 And P() < 80 Then Rating = "Good"
            If P() > 70 And P() > 80 Then Rating = "V.Good"
            If P() > 90 Then Rating = "Excellent"

        End Function
    End Structure





    Sub main()

        Dim S As Student
        Console.Write("Enter Ur Real Name : ")
        S.RealName = Console.ReadLine

        Console.Write("Enter Ur Math Degree : ")
        S.Math = Console.ReadLine

        Console.Write("Enter Ur  Physics : ")
        S.Physics = Console.ReadLine

        Console.Write("Enter Ur Chmestry Degree : ")
        S.Chemestry = Console.ReadLine
```

```
        Console.Write("Enter Ur Piolgy Degree : ")
        S.Piolgy = Console.ReadLine



        Console.WriteLine("UR Total Degree : {0}", S.T)
        Console.WriteLine("Ur Percent Degree : {0}%", S.P)
        Console.WriteLine("Ur Rating: {0}", S.Rating)
        Console.ReadKey()
    End Sub
End Module
```

فى المثال السابق

يعمل نفس عمل المثال الثانى ولكن يزيد عنه فى انه يقوم بحساب التقدير ان كان جيد او جيد جدا او ممتاز اومقبول ... الخ

وايضا يزيد عنه فى انه يستعمل الفانكشن **تابع معى** ...الشرح

وارجو ان تضع كل تركيزك مع الشرح

قمنا بانشاء هيكل بالاسم**Student**

يحتوى على مخزن نصى ليحمل اسم الطالب

و4 مخازن من النوع العددى المزدوج ليحمل مجموع كل ماده

```
Dim RealName As String
Dim Math As Double
Dim Physics As Double
Dim Chemestry As Double
Dim Piolgy As Double
```

اذا اين المخازن التى تحمل المجموع الكلى والنسبه المئويه؟

الاجابه هى : لقد قمنا بانشاء فانكشن يقوم بحساب المجموع الكلى

وفانكشن اخر يقوم بحساب النسبه المئويه

وفانكشن اخر يقوم بحساب تقدير المجموع

اليك التفاصيل

الفانكشن الاول : لحساب المجموع الكلى

```
Function T() As Double
T = Math + Physics + Chemestry + Piolgy
End Function
```

يقوم هذا الفانكشن بجمع مجاميع المواد المحمله فى المخازن لنحصل على المجموع الكلى للطالب

الفانكشن الثانى : لحساب النسبه المئويه

```
Function P() As Double
P = (T() * 100) / 200
End Function
```

يقوم هذا الفانكشن بضرب القيمه الموجوده فى فانكشن المجموع الكلى (المجموع الكلى) فى 100 ثم يقوم بقسمه الناتج على الدرجه العظمى 200
ليعود بقيمه النسبه المئويه للطالب

الفانكشن الثالث : لحساب تقدير الطالب

```
Function Rating() As String
If P() < 50 Then Rating = "Drop"
If P() >= 50 And P() < 60 Then Rating = "Passed"
If P() >= 60 And P() < 80 Then Rating = "Good"
If P() > 70 And P() > 80 Then Rating = "V.Good"
If P() > 90 Then Rating = "Excellent"
End Function
```

يحتوى الفانكشن على مجموعه من الجمل الشرطيه تنص على
لو كانت قيمه فانكشن النسبه المئويه (النسبه المئويه) اصغر من 50% اذا التقدير راسب
لو كانت قيمه فانكشن النسبه المئويه (النسبه المئويه) اكبر من اوتساوى 50% اذا التقدير ناجح
لو كانت قيمه فانكشن النسبه المئويه (النسبه المئويه) اكبر من او تساوى 60% اذا التقدير جيد
لو كانت قيمه فانكشن النسبه المئويه (النسبه المئويه) اكبر من 70% اذا التقدير جيدجدا
لو كانت قيمه فانكشن النسبه المئويه (النسبه المئويه) اكبر من90 % اذا التقدير ممتاز

هكذا كان الهيكل
ثم قمنا باستعماله تحت الصب الرئيسى
امرنا الكونسول ان يقوم بسؤال الطالب عن ادخال قيم درجات المواد ووضع كل قيمه فى مخزنها فى الهيكل
ليصبح الهيكل محملا بمجموع كل ماده
ليتم عمل كل فانكشن كما شرحنا مسبقا ليحمل كل منهم ناتجه
ثم امرنا الكونسول بعرض محتويات كل فانكشن منهم
ليعرض لنا المجموع الكلى والنسبه المئويه والتقدير للطالب

اخى القارىء الشرح طال ؟ اعرف لاننا قمنا بعمل برنامج كبير لحد معين
من فضلك قم بقراءه الشرح اكثر من مره وتطبيق الاكواد لتحصل على ماتريد
وقم بابتكار الاكواد وعمل افكار مختلفه مثلا فكره عرض الكليات المتاحه عن طريق فانكشن مبنى على جمل شرطيه مثل فانكشن التقديرات وافكار اخرى من صميم مادرسناه فى الوحدات السابقه
والان هيا بنا لنطور المثال اكثر من ذلك

المثال الرابع:-

نفس المثال الثالث ولكن يزيد عنه فى استعمال الصب داخل الهيكل '

```
Module module1

    Structure Student

        Dim RealName As String
        Dim Math As Double
        Dim Physics As Double
        Dim Chemestry As Double
        Dim Piolgy As Double


        Function T() As Double
            T = Math + Physics + Chemestry + Piolgy
        End Function

        Function P() As Double
            P = (T() * 100) / 200
        End Function

        Function Rating() As String
            If P() < 50 Then Rating = "Drop"
            If P() >= 50 And P() < 60 Then Rating = "Passed"
            If P() >= 60 And P() < 80 Then Rating = "Good"
            If P() > 70 And P() > 80 Then Rating = "V.Good"
            If P() > 90 Then Rating = "Excellent"
        End Function

        Sub Display()
            Console.WriteLine("-----------------------------------")
            Console.WriteLine("RealName: {0}", RealName)
            Console.WriteLine("Math: {0}", Math)
            Console.WriteLine("chmestry: {0}", Chemestry)
            Console.WriteLine("physics: {0}", Physics)
            Console.WriteLine("piolgy: {0}", Piolgy)
            Console.WriteLine("Total Degree: {0}", T)
            Console.WriteLine("Percent: {0}%", P)
            Console.WriteLine("Rating: {0}", Rating)
            Console.WriteLine("-----------------------------------")

        End Sub
    End Structure
```

```
    Sub main()

        Dim S As Student

        S.RealName = "Meko"
        S.Math = 20
        S.Physics = 12
        S.Chemestry = 15
        S.Piolgy = 20
        S.Display()
        Console.ReadKey()
    End Sub
End Module
```

فى المثال السابق

قمنا بتطويره الى حد ما عن المثال الثالث الزياده فى هذا المثال اننا قمنا بوضع اكواد العرض فى صب داخل الهيكل
اى بدلا من وضعها فى الصب الرئيسى ... قمنا بانشاء صب فى الهيكل ووضعنا به هذه الاكواد
بحيث يتم النداء عليها من الصب الرئيسى بدلا من كتابتها كلها فى الصب الرئيسى
و هذا يسهل علينا العمل فى تصميم البرامج الضخمه
وخاصه عند وجود فريق عمل كبير.

قمنا بعمل هيكل بالاسم **Student**
ثم قمنا بوضع مخازن به من النوع النصى للاسم والنوع العددى المزدوج لمجموع كل ماده
ثم قمنا بانشاء فانكشن لحساب المجموع الكلى(تم شرحه ف المثال السابق)
ثم قمنا بانشاء فانكشن اخر لحساب النسبه المئويه(تم شرحه ف المثال السابق)
ثم قمنا بانشاء فانكشن اخر لحساب التقدير (تم شرحه ف المثال السابق)

ثم قمنابانشاء صب ليحمل اكواد العرض على شاشه الكونسول **Sub Display()**
يحتوى هذا الصب بداخله اكواد عرض النتائج للطالب على شاشه الكونسول فهو يحمل اكواد تنص على

عمل خط (-----) فى بدايه ونهايه شاشه الكونسول لعرض النتائج بينهم
عرض اسم الطالب على شاشه الكونسول
عرض درجه ماده الرياضه
عرض درجه ماده الكيمياء
عرضه درجه ماده الفيزياء
عرض درجه ماده البيولوجى
عرض المجموع الكلى
عرض النسبه المئويه
عرض التقدير
كما ترى الاكواد

```
Console.WriteLine("----------------------")
Console.WriteLine("RealName: {0}", RealName)
Console.WriteLine("Math: {0}", Math)
Console.WriteLine("chmestry: {0}", Chemestry)
Console.WriteLine("physics: {0}", Physics)
Console.WriteLine("piolgy: {0}", Piolgy)
Console.WriteLine("Total Degree: {0}", T)
```

```
Console.WriteLine("Percent: {0}%", P)
Console.WriteLine("Rating: {0}", Rating)
Console.WriteLine("--------------------")
```

هكذا انتهى الهيكل **End Structure**
نأتى الى كيفيه استعماله

كيف نستعمل الان هذا الهيكل فى الصب الرئيسى ؟
مايبقى لنا فى الصب الرئيسى االا ان نعطى له درجات الطالب واسمه هكذا

```
Dim S As Student
S.RealName = "Meko"
S.Math = 20
S.Physics = 12
S.Chemestry = 15
S.Piolgy = 20
S.Display()
```

اى قمنا بحجز مخزن نسخه من نوع الهيكل
ثم اعطينا له القيم كما ترى الاسم ومجموع المواد
ثم امرناه بان يعرض مافى الصب **Display**
الذى قمنا بانشاءه فى الهيكل

مارايك ان قمنا بالتعمق فى استخدام الاستراكشر اكثر من ذلك؟
نجعله يستورد بياناته من قاعده بيانات للطلبه فى مدرسه معينه لنشعر بقيمه المثال!!
ونجعله يقوم بحساب المجاميع على اساس البيانات المسجله فى قاعده البيانات
وبمجرد اننا نعطى له الاسم يرجع لنا بالمجموع الكلى والنسبه المئويه والتقدير
اظن ان المثال كبير ولذيذ
تابع معى... لنقوم بتطبيقه

المثال الخامس:-

قبل ان نقوم بالبدء فى المثال لابد ان نقوم بانشاء قاعده بيانات
سوف نقوم بالشرح على قاعده بيانات سيكوال سيرفر

قمنا بانشاء قاعده البيانات من النوع سيكوال سيرفر
ثم قمنا بانشاء جدول بها وقمنا بتسجيل بيانات لبعض الطلبه فى مدرسه معينه
وتسجيل درجاتهم فى بعض المواد
كما ترى فى الصوره التاليه :-

CAPTIANMALEK-PC.AGC - dbo.Student

ID	RealName	Math	Chemistry	Physics	Piolgy
1	Meko	20.00	30.00	40.00	50.00
2	Shafee	50.00	50.00	50.00	50.00
3	Nour	50.00	22.50	30.00	40.00
4	Salma	22.00	13.00	15.00	14.50
5	Maram	20.00	22.50	50.00	44.20
6	Eman	50.00	50.00	50.00	45.00
7	Mohamed	30.00	22.50	50.00	41.00
8	Yaser	40.00	30.00	20.00	10.00
9	Amera	22.50	30.00	14.00	38.00

اسم قاعده البيانات **AGC**
اسم الجدول **Student**
فكره المثال
نقوم باستدعاء مكتبه الادو ونقوم بعمل مخزن نسخه منها للاتصال بقاعده البيانات ونسخه للاتصال بالجدول كما قولنا من قبل فى **الوحده السادسه**
ثم نقوم بانشاء الهيكل تماما كما فعلنا فى المثال السابق
ولكن سنزيد عليه بعمل صب يقوم بالبحث فى قاعده البيانات عن الاسم المدخل من جهه المستخدم تابع معى ....

نفس المثال السابق ولكن يزيد عنه انه متصلا بقاعده بيانات سيكوال سيرفر'

```
Module module1
    'مخازن الاتصال بقاعده البيانات
    Dim Cn As New ADODB.Connection
    Dim Rs As New ADODB.Recordset

'بدايه الهيكل
 Structure Student
        'مخازن الاسم ودرجات المواد
        Dim RealName As String
        Dim Math As Double
        Dim Physics As Double
        Dim Chemestry As Double
        Dim Piolgy As Double
```

```
        'فانكشن يرجع بالمجموع الكلى للطالب
        Function T() As Double
            T = Math + Physics + Chemestry + Piolgy
        End Function

        'فانكشن لحساب النسبه المئويه
        Function P() As Double
            P = (T() * 100) / 200
        End Function

        'فانكشن لحساب التقدير
        Function Rating() As String
            If P() < 50 Then Rating = "Drop"
            If P() >= 50 And P() < 60 Then Rating = "Passed"
            If P() >= 60 And P() < 80 Then Rating = "Good"
            If P() > 70 And P() > 80 Then Rating = "V.Good"
            If P() > 90 Then Rating = "Excellent"
        End Function

        'صب لعرض النتائج على شاشه الكونسول
        Sub Display()
            Console.WriteLine("-----------------------------------")
            Console.WriteLine("RealName: {0}", RealName)
            Console.WriteLine("Math: {0}", Math)
            Console.WriteLine("chmestry: {0}", Chemestry)
            Console.WriteLine("physics: {0}", Physics)
            Console.WriteLine("piolgy: {0}", Piolgy)
            Console.WriteLine("Total Degree: {0}", T)
            Console.WriteLine("Percent: {0}%", P)
            Console.WriteLine("Rating: {0}", Rating)
            Console.WriteLine("-----------------------------------")
        End Sub

        'صب اخر يقوم بفتح قاعده البيانات ويقوم بالبحث عن الاسم المدخل لعرض درجات المواد
        Sub Find(ByVal Rname As String)
            Cn.Open("provider= SQLOLEDB;Initial 
Catalog=AGC;UID=Sa;Pwd=0124330708;Server=(Local)")
            Rs.Open("Student", Cn, ADODB.CursorTypeEnum.adOpenDynamic, 
ADODB.LockTypeEnum.adLockBatchOptimistic)

            Rs.Find("RealName='" & Rname & " '")
            If Rs.EOF Or Rs.BOF Then
                Console.WriteLine("Name Not Found")
                Exit Sub
            End If
            RealName = Rs("RealName").Value
            Math = Rs("Math").Value
            Physics = Rs("physics").Value
            Chemestry = Rs("chemistry").Value
            Piolgy = Rs("piolgy").Value
            Display()
           Rs.Close()
           Cn.Close()

        End Sub
    End Structure
```

```
    Sub main()

        Dim S As Student
        Console.Write("Enter UR Name : ")
        Dim A As String = Console.ReadLine
        S.Find(A)


        Console.ReadKey()
    End Sub
End Module
```

فى المثال السابق

اولا قمنا باستدعاء مكتبه الادو كما شرحنا مسبقا فى الوحده السادسه لكى نستطيع ان نجعل الكونسول متصل مع قاعده البيانات.

ثم قمنا بحجز مخزن نسخه من مكتبه الادو من النوع كونكشن فى منطقه التصريحات العامه ليقوم بالتوصيل بقاعده البيانات

ومخزن اخر ليقوم بفتح الجدول الخاص ببيانات الطلبه

كل هذا قبل الهيكل

ثم قمنا بانشاء الهيكل

يحتوى على مخازن من النوع العددى المزدوج والنصى لتحمل بيانات الطالب من حيث اسمه ودرجات مواده (تم شرحه فى المثال الثالث)

ثم قمنا بانشا فانكشن يقوم بحساب المجموع الكلى (تم شرحه فى المثال الثالث)

ثم قمنا بانشاء فانكشن يقوم بحساب النسبه المئويه (تم شرحه فى المثال الثالث)

ثم قمنا بانشاء فانكشن يقوم بحساب التقدير (تم شرحه فى المثال الثالث)

ثم قمنا بانشاء صب يقوم بعرض النتائج على شاشه الكونسول (تم شرحه فى المثال الرابع)

ثم قمنا بانشاء صب اخر `Sub Find(ByVal Rname As String)`

هذا الصب مثال الفانكشن يحتوى على معامل نصى يعود يقوم باستقبال قيمه معينه (اسم الطالب)

تابع معى شرح اكواد هذا الصب

اولا يقوم بالاتصال بقاعده البيانات

ثم يقوم بفتح الجدول

ثم يقوم بعمل بحث داخل الجدول عن القيمه المحمله فى المعامل النصى(اسم الطالب)

`ByVal Rname As String`

فان لم يفلح البحث فى ايجاد الاسم المدخل له يعطى رساله على شاشه الكونسول تفيد بان الاسم غير موجود

`Console.WriteLine("Name Not Found")`

واذا وجدها يقوم بعرض ما بكل حقل من حقولها

اى يضع درجات كل ماده من المواد الخاصه بالطالب

فى المخزن المخصص لها فى الهيكل لتصبح المخازن محمله بمجموع درجات كل ماده من المواد

ثم يقوم بالنداء على الصب `Display`

الذى يقوم بعرض النتائج على الكونسول

ثم يغلق الاتصال بقاعده البيانات والجدول

هكذا قد قمنا بانشاء الهيكل

يتبقى لنا كيفيه استعماله!!
سوف ترى سهوله استعماله وقله الاكواد فى استعماله فى الصب الرئيسى
حتى انك ستشعر فى هذا المثال باهميه الهيكله جدا!!!!!!!!! تابع معى ....

تحت الصب الرئيسى
قمنا بحجز مخزن نسخه من الهيكل

ثم امرنا الكونسول ان يطلب من المستخدم ان يدخل الاسم المراد عرض نتائجه من قاعده البيانات
ثم قمنا بحجز مخزن نصى وامرناه ان يحمل القيمه المدخله من المستخدم
ثم بعثنا مافى هذا المخزن
الى الصب `Find` الذى بالهيكل ليقوم بعمل بحث فى قاعده البيانت عن القيمه
ثم يرسلها لباقى الفانكشنات ليقوم كلا منها بعمله

ملحوظه هامه!!!
- **قمنا بحجز المخازن** (التى تقوم بالاتصال بقاعده البيانات والجدول) فى منطقه التصريحات العامه لتعميمها واستعمال ذلك المخازن تحت اى صب واى مكان فى المشروع

-فى الصب `Find`
قمنا باغلاق الجدول اولا `Rs.Close`
ثم قمنا باغلاق قاعده البيانات `Cn.Close`
وذلك لانه لايجوز تقديم قفل قاعده البيانات على قفل الجدول لانك اذا غلقت القاعده البيانات اصبح الكونسول غير متصل بالقاعده اذا لا يستطيع التعامل معها ابدا!!! .
اخى القارىء قمت بالتعليق على ذلك النقطه لانها من النقاط الهامه والدقيقه التى يقع الكثيرون من المبرمجين بها بدون ان يضعو لها بال والسبب الرئيسى ان اعلق عليها انى حين كنت اشرحها اخطأت بها فلذلك كان لابد ان احذرك منها

انتهت الوحده الحاديه عشر

الوحده الثانيه عشر:-

API SPEECH

فى هذه الوحده:-

سوف نتعلم كيف نجعل الكمبيوتر يتحدث بمانريد
كما ترى فى بعض برامج الترجمه يتحدث البرنامج بالكلمه المكتوبه وانت تتخيل انها كلها عباره عن تسجيلات مدمجه مع البرنامج.
والحقيقه ان البرنامج يستخدم الداله API Speech
فانه من الصعب ان يدمج مقطع صوتى لكل كلمات الانجلش مع البرنامج .....دعنا من الحديث هيابنا لنتعلم كيفيه استخدامها ...

المثال الاول:-

يوضح كيفيه جعل الكمبيوتر يتحدث بكلمات معينه'

```
Module Module1

    Sub Main()

        Dim Sp As Object
        Sp = CreateObject("SApI.Spvoice")
        Sp.Speak("I Love You AGC")

    End Sub

End Module
```

فى المثال السابق
قمنا بحجز مخزن كأداه لنستدعى بهاداه SApI.Spvoice
ثم امرنا هذا المخزن ان ينشأ هذه الاداه
Sp = CreateObject("SApI.Spvoice")

من المعروف ان هذه الاداه فائدتها جعل الكمبيوتر يتحدث
اذا اصبح المخزن SP
نسخه من هذه الاداه
لذلك امرناه ان يتحدث بجمله معينه
Sp.Speak("I Love You AGC")

اذا جعلنا الجهاز الان يتحدث بجمله ثابته
فما رايك ان جعلناه يتحدث بكلمه نأمره بها ؟
تابع معى....

المثال الثانى:-

مثال يوضح كيف نجعل الكمبيوتر يتحدث بما نأمره'

```
Module Module1

    Sub main()
        Dim Sp As Object
        Sp = CreateObject("SApI.Spvoice")
        Console.Write("Enter The Words U Like To Speech: ")
        Dim A As String = Console.ReadLine
        Sp.Speak(A)
    End Sub

End Module
```

فى المثال السابق

قمنا بحجز مخزن كأداه لنستدعى بهاداه SApI.Spvoice

ثم امرنا هذا المخزن ان ينشأ هذه الاداه

`Sp = CreateObject("SApI.Spvoice")`

من المعروف ان هذه الاداه فائدتها جعل الكمبيوتر يتحدث

اذا اصبح المخزن SP

نسخه من هذه الاداه

ثم امرنا الكونسول ان يطلب من المستخدم كتابه الجمله التى يريد ان يتحدب بها الكمبيوتر

`Console.Write("Enter The Words U Like To Speech: ")`

ثم قمنا بحجز مخزن من النوع النصى وامرناه بحمل الجمله التى ادخلها المستخدم

ليصبح المخزن النصى محملا بالجمله التى ادخلها المستخدم

لذلك امرنا المخزنSp

ان يتحدث بما يحمله المخزن النصى

```
Dim A As String = Console.ReadLine
Sp.Speak(A)
```

قم بتطبيق المثال اولا

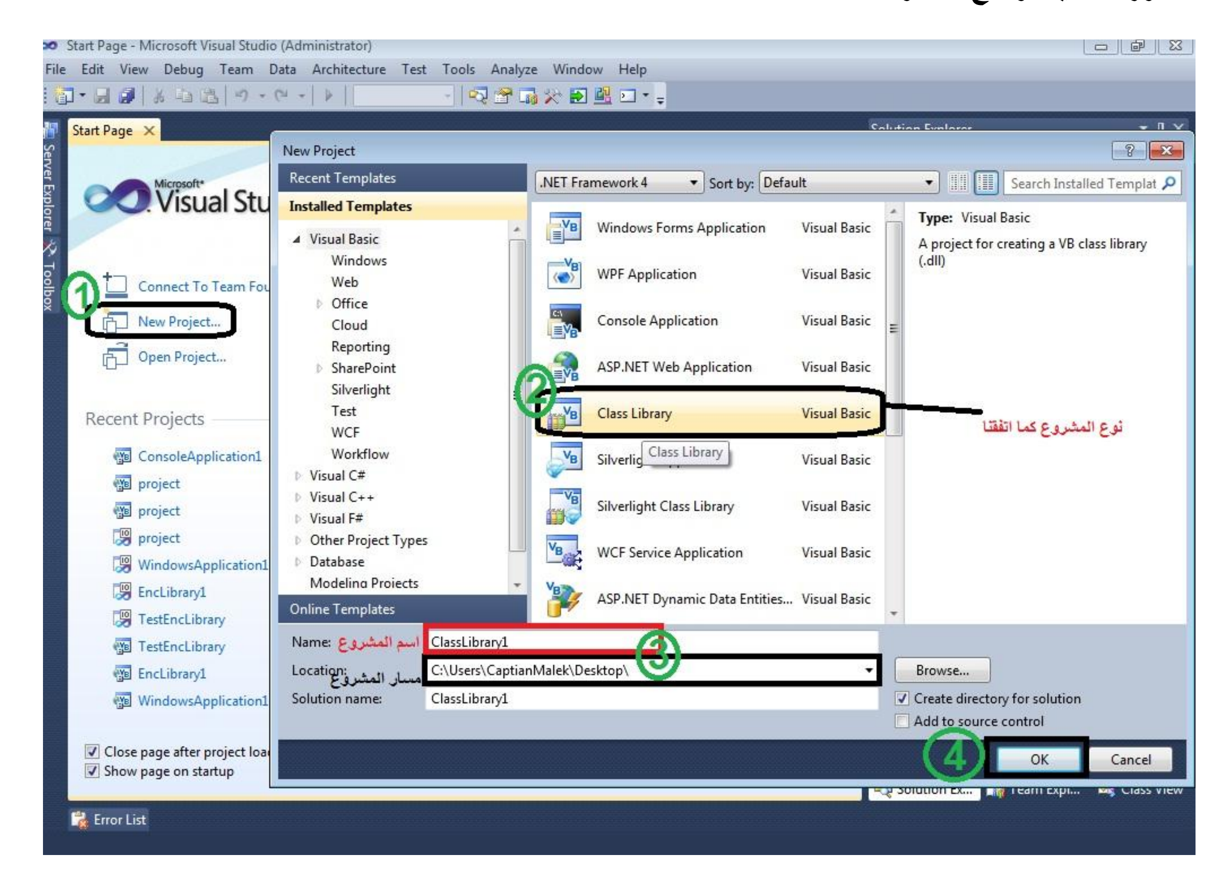
ثم تابعى معى تطويره اكثره بطريقه الـذ

المثال الثالث:-

فى هذه المثال
سوف نتعلم كيف نقوم بعمل مكتبه **DLL**
تقوم هذه المكتبه بقرائه النصوص وسوف نتعلم كيفيه استخدام المكتبه
وما هى اهميتها فى مجموعات العمل

اولا لعمل مكتبه من النوع Dll
نقوم بفتح الفيجوال .نت ثم نختار نوع المشروع ClassLibrary

الصوره التاليه توضح الخطوات



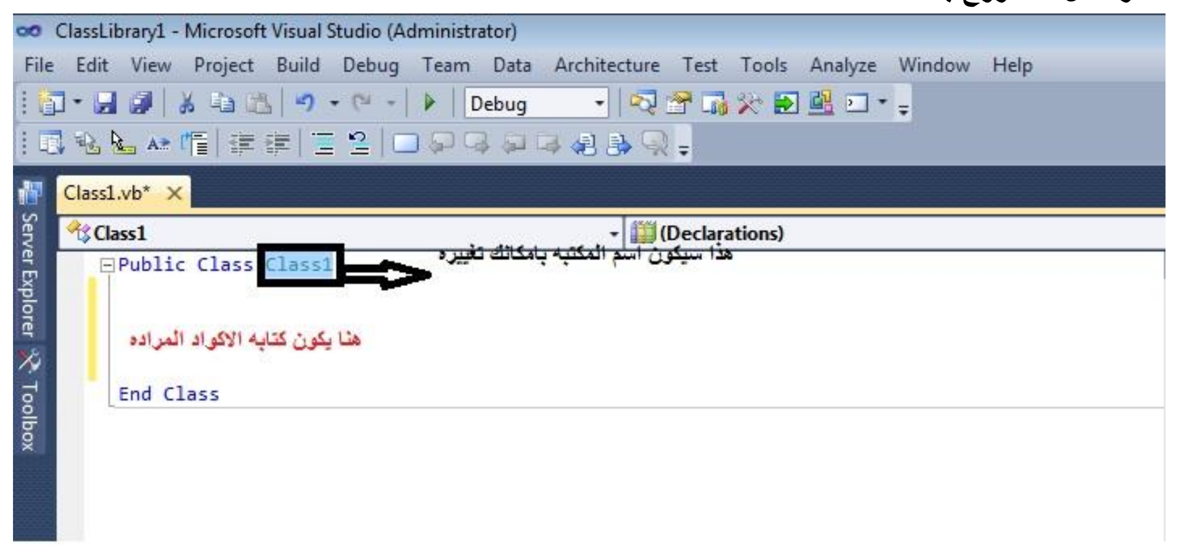
الخطوه الاولى : قمنا بفتح مشروع جديد

الخطوه الثانيه : نختار نوع المشروع

الخطوه الثالثه : نحدد اسم ومسار المشروع

الخطوه الرابعه : نقوم بالموافقه (Ok)وفتح المشروع

انظر شكل المشروع بعد فتحه



عند فتح المشروع نجد Public Class Class1

فان Class1 سيكون اسم المكتبه فيمكن لنا ان نقوم بتغير هذا الاسم ونضع الاسم المراد

هيا بنا الان نقوم بكتابه الاكواد وتغير اسم المكتبه

مثال يوضح كيفيه عمل مكتبه

```
Public Class SpeekVoice

    Sub Speak(ByVal Msg As String)
        Dim Sp As Object
        Sp = CreateObject("SApI.Spvoice")
        Sp.Speak(Msg)
    End Sub

End Class
```

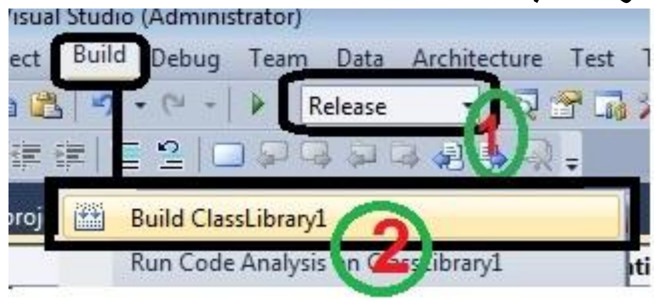
فى المثال السابق

قمنا بتغيير اسم المكتبه الى SpeekVoice
ثم قمنا بعمل صب بالاسم Speak يحتوى على معامل نصى بالاسم Msg
ليستقبل القيمه التى ادخلها المستخدم
ثم قومنا بعمل باقى كود التحدث كما شرحنا سابقا

هكذا قد قمنا بالانتهاء من عمل المكتبه
يتبقى لنا ان نستخرجها من الفيجوال تابع معى...

نضع المكتبه اولا فى الـ Relase من قائمه الادوات
ثم من القائمه الرئيسيه للفيجوال نختار
Build >> Build Class Library1

هذه صوره توضيحيه للخطوات السابقه



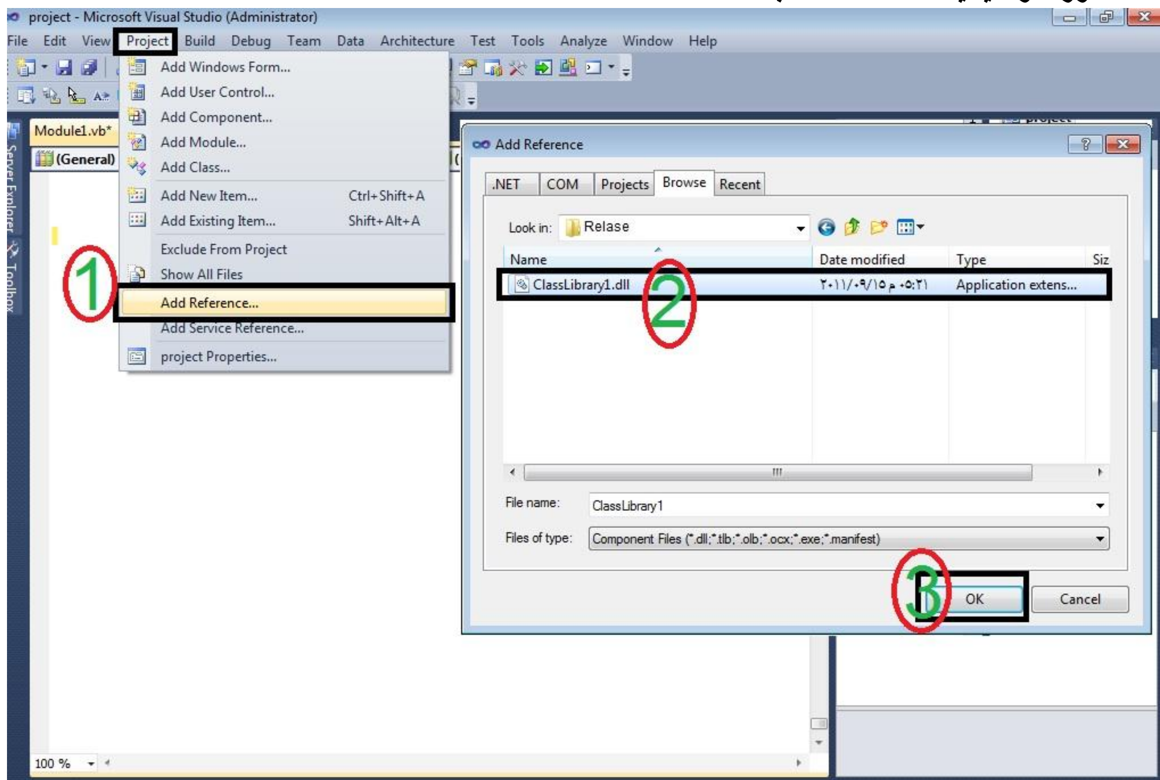
الان قد قمنا ببناء المكتبه بنجاح ونستطيع استعمالها تحت الفيجوال فى اى وقت
ومكان المكتبه الان اصبح فى مجلد المشروع فى المجلد Binثم داخل المجلد Relase
انظر الصوره التاليه



تعالى بنا نتعرف على كيفيه استخدامها مع اى مشروع جديد
تابع معى...

نفتح اى مشروع جديد باى نوع نريد
ثم نقوم اولا باستدعاء المكتبه وذلك من خلال القائمه الرئيسيه
Project>> AddRefrence >> Browse >>نحدد مكان المكتبه ونختار ها

هذه صوره توضيحيه لاستدعاء المكتبه



بعد ما استدعينا المكتبه يتبقى لنا كيفيه استخدامها بالاكواد

تابع ....

```
Sub Main()
        Dim M As New ClassLibrary1.SpeekVoice
        Console.Write("Enter Your Words ")
        Dim N As String = Console.ReadLine
        M.Speak(N)
    End Sub
```

فى الكود السابق

بعدما قمنا باستدعاء المكتبه

قمنا بحجز مخزن M نسخه من المكتبه الى قمنا بانشائها سابقا

Dim M As New ClassLibrary1.SpeekVoice

فان ClassLibrary1 هو اسم مشروع المكتبه السابقه لم اقم بتغيره

و SpeekVoice هو اسم المكتبه نفسها الذى قمنا بتغييره سويا

بعدما قمنا بحجز مخزن نسخه من المكتبه

امرنا الكونسول ان يطلب من المستخدم ان يكتب الكلمات المراد ان يتحدث بها الكمبيوتر

Console.Write("Enter Your Words ")

ثم قمنا بانشاء مخزن نصى وامرناه بحمل القيمه المدخله

Dim N As String = Console.ReadLine

ثم امرناه ان يرسلها للصب Speakالذى انشئناه فى المكتبه سابقا
ليرسل القيمه الى معامله Msg
فيتحدث بها

هذه كانت فكره المكتبه
وللعلم انها من افضل الطرق التى يتعامل بها مجموعات العمل
ليعمل كل شخص فى مجموعه العمل على انشاء مكتبه تقوم بعمل جزء معين من برنامج
والاخر يقوم بانشاء مكتبه اخرى تقوم بعمل حزء معين من البرنامج
ثم يتم تجميع ذلك المكتبات فى البرنامج
ويتكون المشروع

انتهى الجزء الاول من الكتاب

EASY BOOK IN FRAMEWORK

المشاريع المطلوبه :-

المشروع الاول :
تصميم برنامج تحت الدوس يقوم بتشفير الرسائل وفكها
وتدمير الملفات واعادتها لطبيعتها

المشروع الثانى :-
تصميم برنامج يقوم بالاتصال بقاعده البيانات ويكون هو المتحكم الوحيد فى هذه القاعده اى يقوم بتشفيرها
وان يقوم بحساب درجات طلاب الثانويه العامه والنسبه المئويه لهم والتقدير العام وايضا الكليات المتاحه

المشروع الثالث:
انشاء مكتبه تقوم بأى وظيفه معينه
لك الاختيار فى وظائف هذه المكتبه

ايضا ان خطر ببالك فكره اى مشروع اخرى قم بمراسلتنا عبر الايميل لتقييم المشروع
Captian.Malek@Gmail.com

اعلم اخى القارىء
انك اذا قمت بتطبيق المشاريع المطلوبه منك اعلاه
انك فهمت الكورس فهما جيدا
وفى حاله وجود اى عائق فى تطبيق مثال معين
او كود معين قم بمراسلتنا ايضا عبر الايميل وسوف نجيبك على كل الاستفسارات ...

فى انتظار تقيماتكم للكتاب عبر الايميل لنبدأ فى الجزء الثانى

مع تمنايتى لكم بالتوفيق والنجاح

قام بكتابه وتأليف الكتاب
المهندس/مالك صابر محمد

قام بالمراجعه والتصحيح
الاستاذ/ عبدالله صابر محمد